공학박사 학위논문

Content Discovery and Data Offloading in Information-Centric Networking

정보 중심 네트워킹에서의 콘텐트 탐색 및 데이터 오프로딩

2015년 8월

서울대학교 대학원
전기 · 컴퓨터공학부
이 문 영

Content Discovery and Data Offloading in Information-Centric Networking

지도교수 권태경

이 논문을 공학박사 학위논문으로 제출함
2015년 5월

서울대학교 대학원
전기 · 컴퓨터공학부
이 문 영

이 문 영의 박사 학위논문을 인준함
2015년 6월

위 원 장	김 종 권	(인)
부위원장	권 태 경	(인)
위 원	박 세 웅	(인)
위 원	전 화 숙	(인)
위 원	염 익 준	(인)

Abstract

Content Discovery and Data Offloading in Information-Centric Networking

Munyoung Lee
Department of Electrical Engineering & Computer Science
The Graduate School
Seoul National University

While the architecture of current Internet was designed based on the host-to-host communication paradigm for resource sharing, today's Internet usage has been concentrated on content retrievals. As a result, most of Internet traffic is generated by the content retrievals, such as video service and P2P file sharing. However, the discrepancy between the current Internet architecture and the real usage pattern causes inefficient content deliveries (e.g., duplicated content transmission for the same popular content), which leads to traffic explosion problem. To address such issues, there have been a lot of efforts to reduce the network traffic by (i) redesigning the Internet architecture and (ii) proposing data offloading schemes. In this dissertation, we investigate traffic reduction schemes in two different domains, information-centric networking and information-centric vehicular networks.

First, we propose a traffic-reduction content-discovery scheme for information-centric networking (ICN). ICN has been proposed to resolve the problem of current

Internet such as traffic explosion by redesigning the Internet architecture in a clean-slate manner. ICN can provide substantial benefits such as network traffic reduction by exploiting a nearby (cached) copy of content and reducing duplicated transmissions for the same content request. However, prior studies usually rely on an opportunistic cache-hit (happen-to-meet) to utilize the cached contents. In the happen-to-meet fashion, only the content cached on the path towards the content source can be utilized, which limits the network-wide usage of the in-network storages. To exploit cached contents better, our proposed scheme SCAN discovers nearby content copies. SCAN exchanges the cached content information among the neighbor routers using Bloom filters for the content discovery. With extensive simulations, SCAN shows better performance than a happen-to-meet cache-hit scheme in terms of average hop counts, traffic volume, and load balancing among links.

Next, we propose a traffic offloading scheme for information-centric vehicular network. In wireless environments, the increasing mobile traffic is becoming a serious concern for mobile network providers. To address the traffic explosion problem, there have been a lot of efforts to offload the traffic from cellular networks to other networks, such as WiFi hotspots and femtocells. Our work moves forward from prior studies by focusing on the potential benefits of vehicular networks for data offloading. In particular, we propose a Data Offloading framework using Vehicular nEtworks (DOVE), which reduces the cellular traffic for in-vehicle data services in a cost effective way. DOVE exploits vehicle trajectories for offloading purposes so that content files requested by vehicles can be delivered via vehicular networks rather than cellular networks for economical purposes. We formulate the problem of selecting offloading positions as a spatio-temporal set-covering problem, and propose a time-prediction

based set-covering algorithm using vehicle trajectories. Simulation results show that our DOVE framework can significantly reduce 57% of cellular link usage by performing data offloading through vehicular networks.

Keywords : Information-Centric Networking, Information-Centric Vehicular Network, Traffic Reduction, Content Discovery, Data Offloading
Student Number : 2007-21039

Contents

List of Figures

List of Tables

Chapter 1

Introduction

While the architecture of current Internet was designed based on the host-to-host communication paradigm for resource sharing, today's Internet usage has been concentrated on content retrievals. As a result, a recent measurement study [1] shows that most of Internet traffic is generated by the content retrieval applications, such as video service and P2P file sharing. However, the discrepancy between the design of current Internet architecture and the real usage pattern causes inefficient content deliveries. For instance, the same content file is delivered several times inefficiently when multiple users repeatedly request the same content file (e.g., popular contents). Also, a user may retrieve a content file from a distant server even when there is a content copy nearby. These inefficient content deliveries lead to the traffic explosion problem since duplicated content transmissions and content retrieval from a faraway server generate a significant amount of network traffic. In particular, this dissertation focuses on the traffic explosion issue in two different domains; Information-centric networking (ICN) and information-centric vehicular networks. We then investigate two traffic reduction schemes. One is a traffic-reduction content-discovery scheme for ICN and the other is a traffic offloading scheme for information-centric vehicular networks.

First, there have been a lot of efforts to redesign the future Internet architecture in a clean-slate manner [2–8] in order to resolve the inefficient content deliveries and traffic explosion problem. Such a novel networking paradigm called an information-centric networking (ICN) has gained the attention of the research community. In ICN, the networking architecture is designed to be consistent with the current Internet usage pattern (focusing on the contents). By using named contents as identifiers rather than host identifiers to address and retrieve the content, ICN can support efficient content deliveries. Also, contents are cached in in-network storages[1] attached to routers in order to deal with later repeated requests. By exploiting a nearby cached copy of content in the in-network storages and reducing duplicated transmissions for the same content request, ICN provides substantial benefits such as network traffic reduction.

However, existing research on ICN is at an early stage and proposed schemes can be improved. For example, prior studies [3,4,9–12] usually rely on an opportunistic cache-hit (happen-to-meet) to exploit the cached contents. In the happen-to-meet fashion, only the content cached on the path towards the content source can be utilized. However, when considering the portion of in-network storages on the path and the spatial diversity of content requests, the opportunistic cache-hit limits the potential advantages of network-wide in-network storages and the benefits of ICN such as traffic reduction. To fully exploit the in-network storages, we propose a traffic-reduction content-discovery scheme called SCAN that can utilize the network-wide cached contents [13].

Next, information-centric vehicular networks have emerged as one of the promising research areas to support content deliveries for not only drivers/passengers but also software update for car system. The initial research on vehicular networks have focused on the driving safety such as vehicle collision avoidance [14–16]. As cellu-

[1] We use the terms cache and in-network storage interchangeably in this dissertation.

lar communication modules are embedded into cars for in-vehicle data services [17], information-centric vehicular networks have recently drawn the attention of the research community. Through the cellular communications, drivers and passengers can be provided with various in-vehicle data services along with the driving safety. For instance, users can enjoy *delay-insensitive* in-vehicle data services, such as software update for car system and rich site summary (RSS) services (as non-real time applications) for car dash screen (e.g., news headlines and audio/video clips for entertainment). These delay-insensitive services can be available to drivers and passengers via non-cellular links with some delay to reduce the cost of cellular link usage.

However, the increasing mobile traffic is becoming a serious concern for mobile network providers. Considering the current trend of traffic explosion in mobile environments [1], if in-vehicle data services using cellular networks become prevalent, it will significantly worsen the problem of cellular traffic explosion [18]. To address such issue, we propose a traffic offloading scheme called DOVE [19].

The remainder of this dissertation is organized as follows. In Chapter 2, we propose a traffic-reduction content-discovery scheme for ICN. In Chapter 3, traffic offloading scheme for information-centric vehicular networks is presented. Finally, we conclude this dissertation and suggest future work in chapter 4.

Chapter 2

Content Discovery for Information-Centric Networking

2.1 Introduction

This chapter investigates a traffic-reduction content-discovery scheme for ICN in order to exploit cached contents better. To design effective content-discovery scheme for ICN, the following issues should be addressed:

- **How can the information of cached contents be managed?** To discover and retrieve content from a source (i.e., a publisher or in-network storages) at the networking level, the content name information should be maintained in a routing table. However, maintaining the information of all contents in the network is not feasible in terms of scalability [20, 21], since the number of contents in the network is enormous. In particular, if non-aggregatable flat names are employed [2], they highly increase the size of the routing table. Even with the hierarchical names [3, 4], the routing scalability problem still remains when considering not only original contents by the publishers, but also multiple cached copies of each original content in the network. Therefore, the cached content information should be efficiently managed to not significantly increase the routing table size.

- **How can the information exchange overhead be reduced?** Since cached contents are volatile due to the capacity limitation of the in-network storage, the cached content information is periodically exchanged among the content routers (C-routers) for cache consistency. Considering the large number of cached contents and their dispersed names, the traffic overhead for information exchange cannot be negligible. Thus, a compression scheme to reduce the information is needed in order to lower the exchange overhead among the C-routers.

By considering the above issues, we propose a traffic-reduction content-discovery scheme for ICN, dubbed SCAN. To address the first issue, SCAN is designed to manage the permanent original contents and temporary cached contents differently: the information for 'all' the original contents is kept in the routing table[1], while the information for only the 'subset' of the cached contents is maintained in another table. Using the named data networking (NDN)/content centric networking (CCN) as a reference model for ICN, a cache information base (CIB) for cached contents can be populated in addition to the forwarding information base (FIB) for the original contents. In other words, a routing table (i.e., FIB) indicates the next-hop information for the publishers while a separate CIB is maintained for each interface at a C-router to keep the routing information to the network-wide cached contents. Additionally, the second issue is solved when SCAN stores the content information in the CIB and compresses CIBs using the Bloom filter (BF) to keep the table size small and reduce the overhead of information exchange. Note that we assume C-routers cache the contents in their attached storage module as with [2–5, 22].

[1] By exploiting hierarchical name structures, the routing table entries can be aggregated.

By referencing CIBs, SCAN can potentially find multiple close copies of the requested content (we call this operation *"scanning"*). However, there might be a reachability problem if the content discovery process relies only on the CIBs to find the content since the cached contents in the in-network storage can be replaced. To guarantee the reachability with a fallback mechanism in SCAN, a content request is forwarded to the publisher by looking up the FIBs, which is in parallel to the *"scanning"* operation that follows the CIBs. In summary, SCAN improves efficiency in content discovery by the scanning operation while guaranteeing the reachability with the name-based routing of NDN.

The main contributions of this work can be summarized as follows:

- We propose a content discovery scheme 'SCAN' to discover off-path content replicas. To handle a large number of cached contents with small overhead for information exchange, we adopt Bloom filters (BFs) as CIBs and propose a BF-based CIB exchange mechanism using information decay.

- In ICN, a content request (i.e., an interest packet) cannot be sent to a specific location since the location information is not used in the routing. Thus, we devise a content discovery mechanism that can find nearby content sources without requiring the location information.

The remainder of this chapter is organized as follows: In Chapter 2.2, we survey related work and compare SCAN to other proposals. Chapters 2.3 and 2.4 describe the operation of SCAN and the CIB maintenance issues, respectively. Chapter 2.5 presents the evaluation results.

2.2 Related Work

To solve the inefficiency of content retrieval in the current Internet architecture, there have been several efforts to redesign it in a clean-slate manner. Named data networking (NDN) [3, 4] is a representative architecture based on a content-centric paradigm. NDN adopts a hierarchical name structure to aggregate content names for name-based routing. Since SCAN chooses NDN as a reference ICN model and integrates its content discovery scheme with NDN, we will briefly introduce the operation of NDN. However, note that SCAN can coexist with various ICN models since SCAN leverages the underlying ICN routing model (main routing scheme) and adds the scanning operation as a subsidiary mechanism.

2.2.1 Named Data Networking (NDN)

NDN uses an interest packet and a data packet for content request and retrieval. The interest packet contains the requested content information while the data packet is used to deliver the requested content. The sketchy operation is as follows: (1) an end host sends an interest packet with the name of the requested content in the chunk unit. Then, the interest packet is forwarded to the content publisher; (2) when a C-router receives the interest packet, it looks up its content store (CS) table to check whether the requested content is cached or not. If cached, the C-router sends the data packet to the soliciting host. Otherwise, the interest packet is forwarded to the next interface based on the FIB. After forwarding the interest packet, the forwarding history is recorded in the pending interest table (PIT) to remember the backward path and reduce duplicate request forwarding; (3) during data delivery, the C-routers on the backward path can store the data for later requests. SCAN extends NDN to utilize the cached contents more extensively by introducing the scanning operation.

2.2.2 ICN-based Schemes

There have been several ICN architectures [2, 6, 7] that adopt a name resolution system for content retrieval. Data-oriented network architecture (DONA) [2] uses flat and self-certifying names where the content requests are routed over the predefined tree hierarchy of network entities (called resolution handlers (RHs)) to find the closest copy. Similarly, PSIRP [6] and NetInf [7] find a nearby copy by using a name resolution system. However, the scalability problem is a major concern for these approaches since the name resolution system should maintain the list of copies for all contents.

Some studies [3,21–24] introduce schemes that directly utilize a content name to locate the content. These works propose a routing protocol in the control plane. However, to make a forwarding decision, every node should know the content popularity in advance. Eum *et. al.* [21] investigated a secondary best-effort routing mechanism named potential based routing (PBR) to boost availability of copies. In PBR, a user request is attracted by the cached content information called a potential field. Both PBR and SCAN utilize a subsidiary mechanism to discover off-path content replicas. However, PBR manages cached contents without information compression, and a content retrieval path might be stretched when a request follows the potential field of removed cached content. On the other hand, SCAN compresses cached content information using BFs to lower the information exchange overhead, and the worst-case content retrieval path can be bounded by a fallback mechanism. Saino *et. al.* [24]

applied hash-routing techniques to ICN. The hash-routing is a scalable solution to manage cached contents. However, in-network storages cannot be fully utilized since the locations of content copies are decided by a hash function. Wang *et. al.* [23] mentioned the feasibility of using BFs for advertising the cached content information in ICN. In their proposal, link state advertisements are used for information exchange so each interface has multiple BFs to maintain the hierarchical cache information. Thus, to maintain the cache information of n-hop neighbors, a router should exchange the information with n neighbors at each interface. In contrast, SCAN exchanges the information among 1-hop neighbors since the cache information of neighbors is aggregated to a single BF with a decay probability. Therefore, SCAN generates less information exchange overhead than [23].

Also, several works [20,25,26] focus on data plane forwarding strategies. Chiocchetti *et. al.* [20] proposed a hybrid forwarding strategy that combines the flooding mechanism for cached contents and the shortest path forwarding for original contents. If the requested content is popular, flooding is used to utilize a cached content while non-popular content requests are forwarded to the publisher. In [25], authors designed a dynamic forwarding scheme (called INFORM) that discovers available content replicas. INFORM adapts request forwarding based on the delivery time estimations using a reinforcement learning. In [26], Yi *et. al.* introduced an adaptive forwarding scheme in the NDN framework. In their approach, available interfaces are periodically probed to retrieve data via the best path. These works [20, 25, 26]] try to exploit only short term content item availability in the data plane. In contrast, SCAN focuses on the distribution of information about mid-long term item copies in the control plane.

In our preliminary work [22], a content routing scheme based on IP protocols was proposed. In [22], the proposed scheme finds the location information of cached contents, and then retrieves the content from a nearby content source by sending a content request to the target location. Thus, it cannot be applied to the ICN environments (i.e., NDN) directly where location information is not utilized in the routing process.

2.2.3 Approaches using BFs

There have been proposals to utilize BFs in the routing. Scalable query routing (SQR) [27] adopts exponentially decaying BFs to route the queries in unstructured P2P networks. Queries will be forwarded to the neighbor peer that has the highest number of matched bits. A content query initially follows a random walk until it meets a peer that has the routing information of the requested content. In SCAN, the content request will follow both the FIB (name-based routing towards the publisher) and the CIBs (scanning operation for cached contents), which prevents an unnecessary random walk. In [28], Rhea *et. al.* proposed a probabilistic routing algorithm to enhance the performance of IP routing. Attenuated BFs help find the nearby copy with higher probability. Thus, the traditional IP routing is enhanced through a hybrid approach: first try the probabilistic algorithm and then follow the deterministic IP routing if needed. In contrast, SCAN utilizes nearby cached contents in the networks by gradually scanning around the path to the publisher.

2.3 SCAN Architecture

The key feature of SCAN is leveraging both a content discovery operation (i.e., *scanning* operation), to improve efficiency, and NDN routing, to guarantee the reachability. The goal of the scanning operation is to find network-wide cached content copies to enhance the content retrieval while that of NDN routing is to ensure that the content is delivered to the requestor. If the scanning operation cannot find the cached contents, the content retrieval is guaranteed by NDN routing that exploits the en-route path as a fallback.

In this chapter, the operations of SCAN will be described in NDN environments [3, 4]. Hence, our scheme assumes the existence of functional entities such as CS, PITs, and FIBs for the NDN routing.

2.3.1 SCAN Description

We assume the universal caching scenario where all C-routers in the network can make independent caching decisions about which content will be cached inside their storage modules. C-routers perform a content discovery called scanning, where it sends a *scanning request* in addition to an original content request (e.g., an interest packet in NDN). A content request is generated by a host while a scanning request is generated by the C-router that relays the content request. Both requests contain the information of the requested content, and they can simply be distinguished by an additional bit in the packet header.

To locate the requested content, SCAN uses a content name[2] for both scanning

[2]We assume that every content is specified by a content name (e.g., hierarchical name in NDN, an HTTP URI).

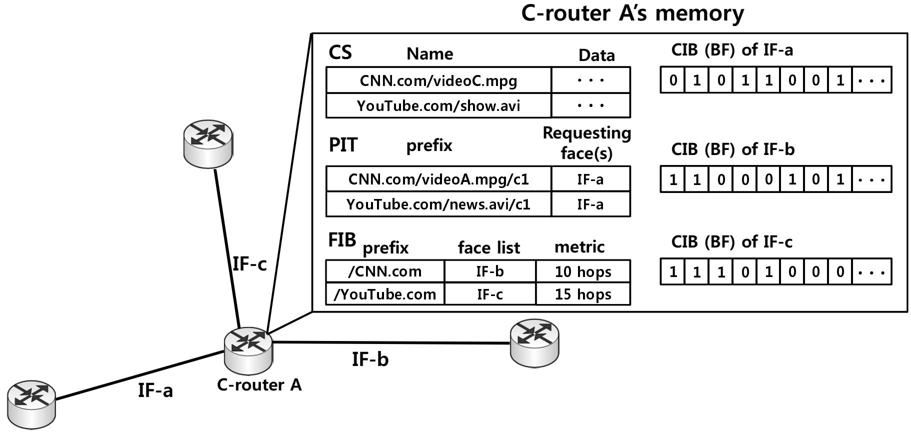


Fig. 2.1. The structure of SCAN.

and NDN routing. The content name can be extracted using the deep packet inspection (DPI) [29] or easily extracted from the content request since the name itself is used for the routing in ICN. Due to the scalability issue, we assume that FIBs maintain the information of hierarchical names (i.e., publisher/content name). Hence, the content request is forwarded to the publisher of the content by looking up the FIBs.

For the SCAN operation, a C-router maintains a CS, PIT, FIB, and CIBs as shown in Fig. 2.1. The PIT and FIB are forwarding entities for the main routing scheme. If SCAN leverages other ICN routing models instead of NDN, the forwarding entities of C-router can be changed. The CS consists of the cached content information at the C-router, and the PIT maintains the forwarding history in order to remember the backward path to the soliciting host. The FIB stores the routing information for the publishers of the original contents. In addition to CS/PIT/FIB, we add CIBs to maintain the routing information for the cached contents in the neighboring C-routers. There is one CIB for each outgoing interface of the C-router, which accumulates the information of cached contents located at neighbor C-routers that are reachable via that interface. Note that the next-hop information for the cached contents found by the scanning operation can be temporarily inserted into the FIB with a short lifetime to enhance the performance. We call this next-hop information 'temporary entry by scanning' (TEBS).

Since cached contents in the in-network storages are changed by a cache replacement policy, each C-router periodically exchanges the cached content information with neighbor C-routers. Considering the large number of cached contents, the table size of each CIB may be so large that the traffic overhead for information exchange cannot be negligible. To reduce the information exchange overhead of CIB, SCAN compresses the content information stored in the CIB by using a Bloom filter (BF)[3] [30]. A BF is a space-efficient random data structure which supports the membership queries for a set of elements. Note that the BF size is fixed. Thus, the complexity of CIB using BF is calculated as the product of the BF size and the number of interfaces of C-router. For instance, if a C-router has 6 interfaces and the BF size is 2,500 bits, each C-router's additional complexity of CIBs is 1,875 bytes. We believe this complexity of the additional data structures is not significant.

A typical BF is embodied as a series of m-bits. When an element α is to be added to the BF, say S, α is hashed by k independent hash functions. Each hash function returns an integer between 0 and $m-1$. Among m bits of S, the bits corresponding to the results of k hash functions are set to 1. One can decide whether S contains α by checking whether the corresponding k bits are set to 1. However, there is also a possibility of a *false positive*. To sufficiently lower the possibility of false positive

[3] In this dissertation, we use the terms CIB and BF interchangeably to indicate the routing table for cached contents.

decisions, the size of a BF, m, should be large enough compared to the number of stored elements, n, and the number of hash functions, k (i.e., $k \times n \ll m$). For the BF management, we can use hash functions such as MD5, CityHash64, Spooky, and Jenkins, which have low hash collision probability [31]. In SCAN, these hash functions are assumed to be managed and predefined within a single domain. Regarding multiple domains, we assume that the BF information (e.g., choice of hash functions) is exchanged through an external protocol (e.g., inter-domain routing protocol).

The key advantage of using BFs is that BFs can condense the membership information into the limited size and check the inclusion of an element quickly. In SCAN, C-routers maintain one BF for each interface and BF compresses the information of the contents that are advertised through the interface. If k hashed values of the name of requested content are matched to the BF of a particular interface, the C-router can conclude that the requested content may be located through the interface at a high probability. In Chapter 2.4, we will explain how CIBs are maintained and exchanged in SCAN.

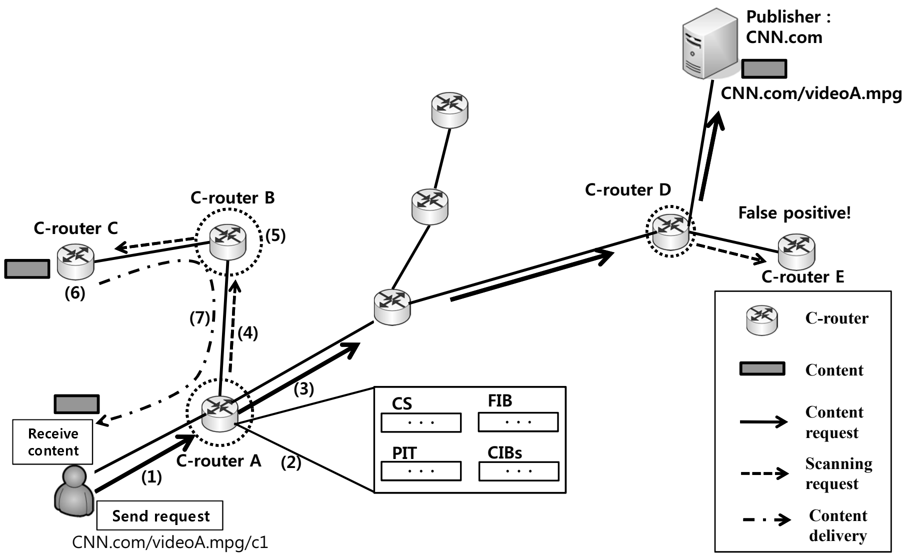


Fig. 2.2. The SCAN operation.

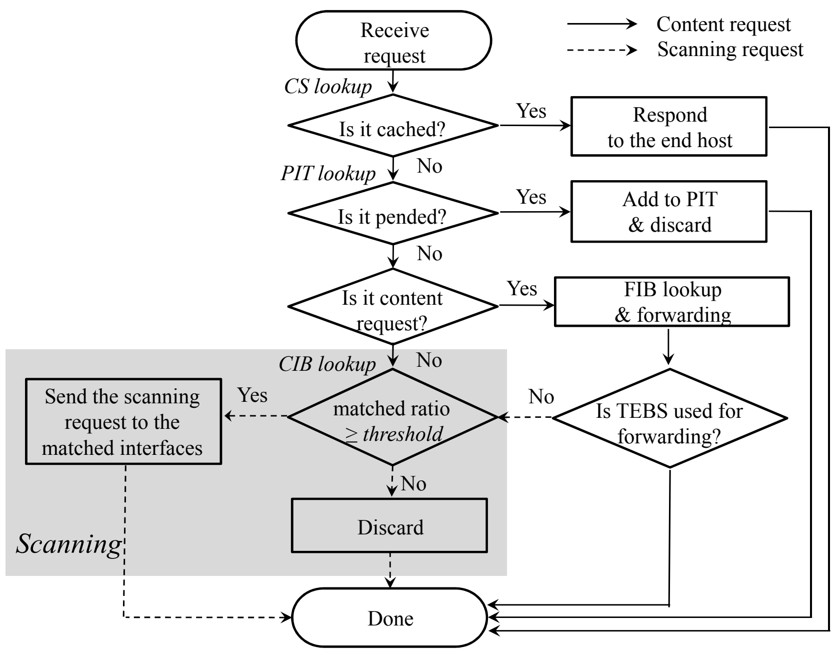


Fig. 2.3. The forwarding decision at C-router.

2.3.2 SCAN Operation

This chapter explains the overall operation of SCAN as illustrated in Fig. 2.2 and Fig. 2.3.

- Step (1): When an end host sends a content request with the requested content name, the content request for each chunk is forwarded towards the publisher of the content.

- Step (2): When a content request arrives at C-router *A* that is on the path to the publisher, C-router *A* makes a forwarding decision as shown in Fig. 2.3. First, C-router *A* checks its CS and PIT in succession to check whether there is matched information or not. If C-router *A* holds the requested content in the CS, it sends the data to the end host (the one that issued the content request). If not, it looks up the PIT to avoid duplicated request forwarding. In Fig. 2.2, a C-router *A* receives the content request and detects there is no matching information in the CS and PIT. Thus, it looks up a FIB to forward the content request to the next interface.

  Since a FIB consists of original FIB entries and temporarily inserted entries by a scanning procedure called TEBS, SCAN performs different operations in steps (3) and (4) according to the type of FIB entry. The original FIB entry indicates the path to the publisher while the TEBS inserted from the previous scanning operation leads to the nearby cached content. In SCAN, TEBS is needed to reduce unnecessary traffic overhead. We will describe this in detail in Chapter 2.3.3.

- Step (3): After obtaining the forwarding information from the FIB, C-router *A* sends the content request to the next-hop interface as shown in Fig. 2.2. Then, C-router *A* checks whether TEBS was used for request forwarding. If TEBS was used for forwarding, C-router *A* does not perform the scanning operation since the content request has already been forwarded to the close cached content. If the forwarding information belongs to the original FIB entry, which means that the scanning operation does not occur, C-router *A* looks up the CIBs to send additional scanning requests.

- Step (4): After forwarding the content request, C-router *A* performs scanning (the gray region in Fig. 2.3). C-router *A* checks CIBs (BFs) associated with every outgoing interface except the one in which the content request comes in. It takes O(1) time to check whether the information of the requested content is within the CIB. If there are matched interfaces (i.e., the number of matched bits is equal to or greater than the predefined threshold), the C-router *A* sends additional scanning requests towards C-router *B* as shown in Fig. 2.2. Note that if there is knowledge (TEBS) in step (3), C-router *A* only sends the content request to C-router *B* instead of forwarding it to the publisher. Next, the content request will be forwarded according to the TEBS to utilize the close cached content.

- Step (5): When a scanning request arrives at C-router *B*, it first looks up its CS to check whether the requested content is cached or not. In Fig. 2.2, C-router *B* does not hold the requested content and no matched information exists in the PIT. Therefore, C-router *B* conducts the scanning operation. It checks CIBs and sends the scanning request to the target interfaces if there are matching interfaces. From this scanning operation, the scanning requests may eventually reach multiple C-routers caching the same requested contents. Note that when a C-router receives the scanning request, it does not perform the NDN routing since its purpose is to find a cached replica.

- Step (6): When C-router *C* caching the requested content receives the scanning request, it sends the content (chunk) to the end host (the one that issued the content request). If there are multiple copies, the cached content located at the nearest C-router will arrive first and redundant data traffic can be minimized due to the PIT (i.e., preceding data already consumed the request). From this operation, SCAN can retrieve the requested content from a nearby C-router.

- Step (7): During data delivery, the requested content can be cached at intermediate C-routers *A* and *B* that are located on the backward path. If data is returned from the matched interface, intermediate C-routers record this information (called TEBS) into the FIB with a short life-time (i.e., soft state) to guide the requests for subsequent chunks. If corresponding TEBS already exists in the FIB, the C-routers update the life-time. The content requests for subsequent chunks can utilize the TEBS from the FIB, and they are forwarded to the nearby C-router without the scanning operation. As the valid time of entry for the cached content is relatively short and updated by returned data in the FIB, the entry of a replaced content will be removed after timeout. Then, requests for subsequent chunks will perform the scanning operation to find a nearby content.

2.3.3 Discussion

Role of probing: As the response to a request (i.e., an interest packet) is data itself in NDN, each scanning request may cause duplicated data traffic from multiple C-routers. Thus, performing scanning operations for every chunk increases extra traffic overhead. To reduce unnecessary overhead, only a small portion of requests can perform the scanning operation. For instance, 1 out of 100 requests for content chunks can trigger the probing of cached contents by sending scanning requests. Then, the scanning result (called TEBS) is recorded to FIBs for the following requests. Requests for subsequent chunks can utilize the scanning result to retrieve the closest copy.

Scanning depth: Since retrieving contents cached at distant locations may be inefficient, we need to control the exploration range of the scanning operation, called *scanningDepth*. The *scanningDepth* is defined as $min(\beta \times remainingHops, depthMax)$, the maximum hop count within which the scanning request can traverse in order to find a copy. β is the unit of threshold between 0 and 1, *remainingHops* is the remaining hop count from a current C-router to the publisher[4], and *depthMax* is the predefined maximum value of *scanningDepth*. The rationale behind the *scanningDepth* is to perform the scanning more thoroughly near the requesting host (e.g., *scanningDepth* is large near the host since the remaining hop count is also large). *scanningDepth* decreases by 1 as the scanning request traverses at each C-router, and the scanning operation stops when *scanningDepth* equals 0. The evaluation of scanning operation according to *scanningDepth* will be described in Chapter 2.5.4.

[4]We assume a FIB indicates the number of hops or latency as a *metric* field in the current routing table in order to select the best route.

False Positive: The advantage of using a BF is that a BF can effectively store a large number of content information in a fixed size bit string. In addition, one can easily verify when the content information is included in the BF. However, a false positive may occur due to the nature of a BF, even when a sufficiently large one is used. In general, the false positive in routing schemes prohibits a correct retrieval of the requested content since it causes a reachability problem. As shown in Fig. 2.2, the false positive in SCAN makes the scanning requests (e.g., generated from C-router D) be propagated to irrelevant C-routers (e.g., C-router E), which generates unnecessary control traffic. However, SCAN can mitigate the negative effects of false positives with the fallback mechanism. Because SCAN leverages both the scanning operation for cached contents and the NDN routing for original contents, the requested content can be retrieved from the publisher even if the false positive decision has been made. Therefore, the false positive in SCAN causes additional traffic overhead, but the content reachability is not affected. The evaluation of the false positive effect will be described in Chapter 2.5.6.

Control Overhead: SCAN incurs additional control traffic compared to the pure NDN routing since the CIB information should be exchanged and additional scanning requests are generated to find multiple copies of the requested content. However, SCAN can reduce the total volume of the network traffic as well as inter-AS traffic by utilizing the nearby cached contents, which will be described in Chapter 2.5.3.

2.4 CIB Maintenance in SCAN

To utilize network-wide cached contents, each C-router should share the information of its own cached contents among neighbor C-routers. Since the number of cached contents in the network can be huge and cached contents in in-network storages are dynamically changed by a cache replacement policy, the maintenance of CIBs is an important issue. In this chapter, we will describe how to manage and exchange CIBs in SCAN.

2.4.1 Information Unit

If content information is managed as a chunk-based unit, a large CIB is needed to cover all the chunks of content. Also, the information in each chunk in the CIB is handled as if each chunk is a single cached unit. Therefore, even if sequential chunks are cached at the same C-router at a high probability, the scanning operation for every chunk results in redundant traffic. [5] To reduce unnecessary overhead and *keep a relatively small CIB*, we consider an entire file as a cached unit for information management. Thus, the entire content information is maintained in a CIB and shared among neighbor C-routers. If a preceding request packet (role of probing) discovers the requested content through scanning, request packets for subsequent chunks can utilize this knowledge without additional scanning overhead.

[5] Because interest and data packets are 'one-for-one', a data packet is transmitted only in response to an interest packet [4].

2.4.2 Information Exchange

For the scanning operation, C-routers exchange their CIBs periodically. For example, when C-router *A* and C-router *B* exchange their CIBs, C-router *A* merges BFs of all the interfaces except the interface associated with C-router *B* and sends the merged BF (the information of C-router A's neighbor) to C-router *B*. The cached contents in *A*'s CS will also be inserted into the merged BF. Then, C-router *B* adds the received information to the BF of the interface connected to C-router *A*. Similarly, C-router *B* sends its own CIB to C-router *A*, and C-router *A* will reflect it into the corresponding BF. Since the compressed BF on CIB is exchanged among the neighbor C-routers, the traffic overhead generated by BF exchanges has little impact on the total network traffic volume. We will elaborate on this issue in Chapter 2.5.7.

Note that the content information in the CS can be managed by using a counting bloom filer [30]. The counting bloom filter is an extended version of BF (from a single bit to an n-bit counter) that provides a delete operation. Using the counting bloom filter, the content information in the cache is dynamically compressed and managed when the cache status is changed (content is added or removed). Thus, the time cost of the BF construction for the BF exchanges includes the conversion time of the counting bloom filter to the ordinary BF. With the easy conversion mechanism, the counting bloom filter is converted to the BF. For instance, the value larger than 0 in the counting bloom filter is converted to 1, and the value equal to 0 is converted to 0. It takes O (the number of bits in BF) time for the BF construction, which alleviates the BF exchange overhead.

The validity of the advertised content information depends on the lifetime of cached contents since they are evicted by the cache replacement policy. To make an advertisement meaningful, the advertised cached contents in a CS should remain until the next information exchange. A simple way to utilize an advertisement is to slow down the replacement rate of cached contents through regulating a cache replacement [23]. However, a replacement should occur rather frequently since the caching benefits at packet level happen in the first 10 seconds [32, 33].

Therefore, we adopt a two-level CS to obtain benefits from both the cache advertisement and short-time caching. The two-level cache consists of a temporary space (memory) and a protected space (storage). The temporary space is used to store all the incoming contents that are evicted with the least recently used (LRU) replacement policy for the line speed packet forwarding. The protected space preserves cached contents that are advertised in SCAN until the next information exchange in order to keep the advertised information consistent. Thus, at every information exchange, the current status of the temporary space is copied to the protected space. The cached content information in the protected space is compressed using the BF for the information advertisement. Note that the cached contents in the protected space can be maintained based on the content popularity. However, keeping the information on the content popularity can be a bit of a burden to the C-router.

2.4.3 Information Decay

When a C-router performs the CIBs exchange, SCAN adopts an information decay mechanism. Thus, the C-router decays all the bits (that are set to 1) of the merged BF (the information of C-router's neighbor) probabilistically similar to [27]. For example, if the decay probability is 0.5, the half of the 1 bits are changed to 0 bits. Then, the C-router inserts the information of its own cached contents into the merged BF and exchanges the BF.

The rationale behind the information decay mechanism is as follows: First, we can estimate the distance of C-routers that hold the requested content. Since the original information of a CIB is delivered over multiple hops with probabilistic decay, some information in the CIB (hashed bits in a BF) will disappear as it passes through multi-hop C-routers. Therefore, CIBs can indicate the existence of a cached content more precisely when the C-router holding the content is located nearby (i.e., the *matched ratio*[6] is close to 1). In contrast, only partial information of faraway C-routers can be maintained in BFs. In other words, the information decay mechanism probabilistically eliminates the cached content information that is stored in faraway C-routers to exploit the nearby cached contents better. Second, the information decay mechanism can reduce the probability of false positives. If CIBs, represented BFs, are aggregated over multiple hops, it is possible that all the bits of a BF are set to 1, since the bits for a large number of contents will be accumulated. This will raise false positives significantly, and thus a BF may not be effective. To address this problem, SCAN uses the information decay mechanism to prevent that all the bits of a BF becomes 1 and it can guarantee that the BF can be used as a content routing table.

Due to the information decay, BFs may not include the information of the requested content even if the content is cached in neighbor C-routers (i.e., false negatives). To handle this issue, as shown in Figure 2.3, scanning can be performed based on a BF matching threshold rather than a perfect match when there is no perfectly matching BF. In this case, the scanning request can be forwarded to the matching interface if the *matched ratio* is over the threshold. The BF matching threshold can

[6] matched ratio $= \frac{A}{B}$ where A is the number of matched bits between the BF and hashed values of requested content, and B is the number of hashed bits.

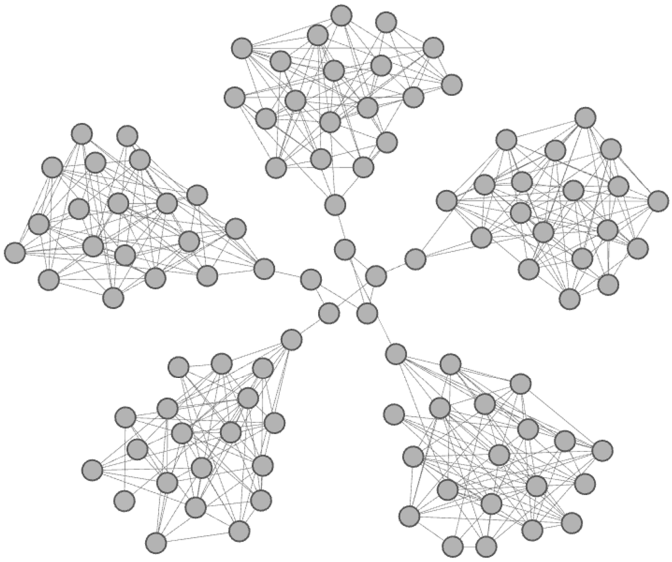


Fig. 2.4. Simulation topology (1 transit domain, 5 stub domains).

be determined by considering the tradeoff between discovery probability and control overhead. If the threshold is close to 0, the probability of false negatives decreases and the number of discovered contents increases due to widely spread scanning requests like flooding, but it generates high control overhead. Conversely, as the threshold is close to 1, low control overhead is expected. However, it may not fully discover cached contents due to restricted forwarding of scanning requests.

2.5 Performance Evaluation

In this chapter, we evaluate the performance of SCAN. We conducted simulations using a discrete event-driven simulator. Table 2.1 summarizes the detailed descriptions of the simulation configuration. As shown in Fig. 2.4, GT-ITM [34] is

Table. 2.1. Simulation Configuration (SCAN)

Parameter	Description
Simulation topology	Internet-like topology which consists of 1 transit domain, 5 stub domains, 105 C-routers.
Number of publishers	The number of publishers is 100. Collocated with the C-routers.
Number of subscribers	The number of subscribers (end hosts) is $1,000$.
Number of contents	The number of contents (Content catalog) is $100,000$.
Size of the content	The size of the content is 1 GB.
Content requests	The probability distribution of content requests follows the Zipf distribution with a parameter of 1.0.
Cache size	The size of in-network storage. The default storage size is 10 GB.
Bloom filter (BF)	The size of BF is 2,500 bits with 10 hash functions (MD5).
Matching threshold	The threshold for BF matching is set to 0.9.
Decay probability	The probability of diminishing the BF bits for information decay. The default decay probability is 0.5.
Scanning depth	The unit value of the scanning depth threshold β is 0.5.

used to generate an Internet-like topology which consists of 1 transit domain, 5 stub domains, 105 C-routers. There are 100 publishers (collocated with the C-routers in stub domains) and 1,000 end hosts. 100,000 contents are randomly distributed among publishers. Hence, each publisher stores an average of 1,000 contents. The size of the content is 1 GB, and each content is divided into 1,000 chunks. The probability distribution of content requests follows the Zipf distribution with a parameter of 1.0 [12]. To cache the content, each C-router has in-network storage with its default storage size set to 10 GB unless otherwise specified. Each BF is set to 2,500 bits long with 10 hash functions and the threshold for BF matching is set to 0.9. Also, the default decay probability is 0.5, which diminishes half of the BF bits for information decay while the unit value of the scanning depth threshold β is set to 0.5.

We compare SCAN with legacy IP routing (denoted by IP), pure NDN (denoted by NDN) [3,4], and SCAN maintaining chunk-based information (denoted by

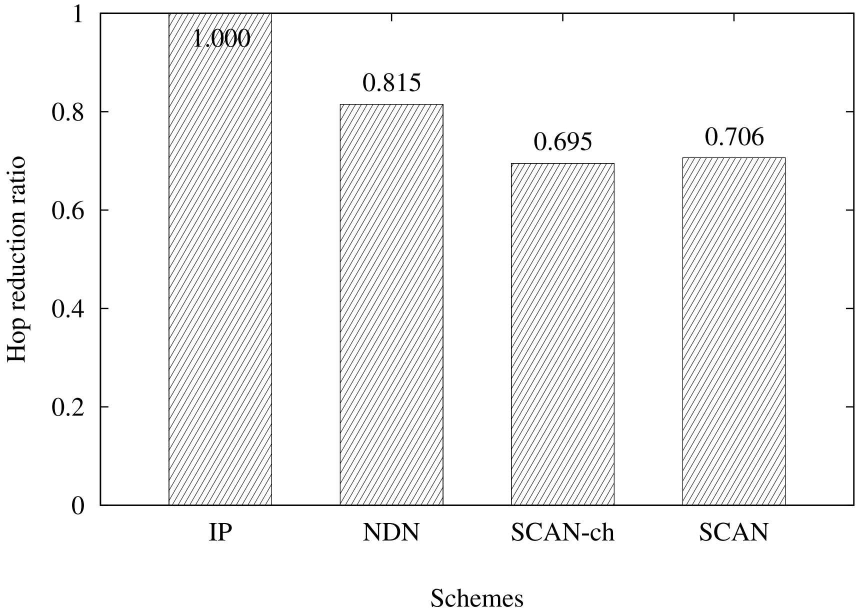


Fig. 2.5. Average hop reduction ratio.

SCAN-ch). In NDN, a C-router can utilize cached contents located at en-route path by happen to meet manner. The operation of SCAN-ch is similar to SCAN except in the information management. Since SCAN-ch uses chunk-based units for information management, a large BF is needed and the scanning operation is performed at every chunk.

2.5.1 Content Discovery Performance

We measure the average hop count[7] to evaluate the performance of content discovery (scanning). As the average hop count decreases, an end host is expected to

[7]We assume ICN environments where only C-routers are deployed. Thus, the hop count means the hop distance between C-routers.

possibly download the content faster from a nearby C-router and the total network traffic can be reduced. To show the gains of the scanning operation, we calculate the hop reduction ratio that is defined as the relative hop count compared to the retrieval from the publisher. As shown in Fig. 2.5, the hop reduction ratio of IP routing is 1, since IP routing that retrieves data from the publisher is a base line. SCAN achieves a small number of hop counts (30% reduction) for content delivery due to the help of content discovery. From Fig. 2.5, it can be seen that SCAN-ch shows smaller hop counts than SCAN. This is because SCAN-ch accelerates content distribution by sending more scanning requests, which leads to the migration of cached contents. In contrast, NDN can achieve about 19% reduction since only the content cached on the path between a host and a content-holding place (either the publisher or the closest C-router that caches the content) can be utilized.

2.5.2 Network-wide Performance

To show the network-wide performance, we measure the average control overhead, which is defined as the traffic volume of control packets (e.g., content request and scanning request). As shown in Fig. 2.6, SCAN-ch generates the highest control overhead since it sends scanning requests (performs scanning operations) for all the content chunks. Furthermore, the control overhead of SCAN is greater than IP routing and NDN. The gap between them represents additional control overhead generated from content discovery. However, as shown in Fig. 2.7, the reduction in data traffic overwhelms the additional control overhead if the discovered content returns from a close C-router. This shows the benefits of SCAN.

To show the load balancing and mitigation of network traffic, we measure the

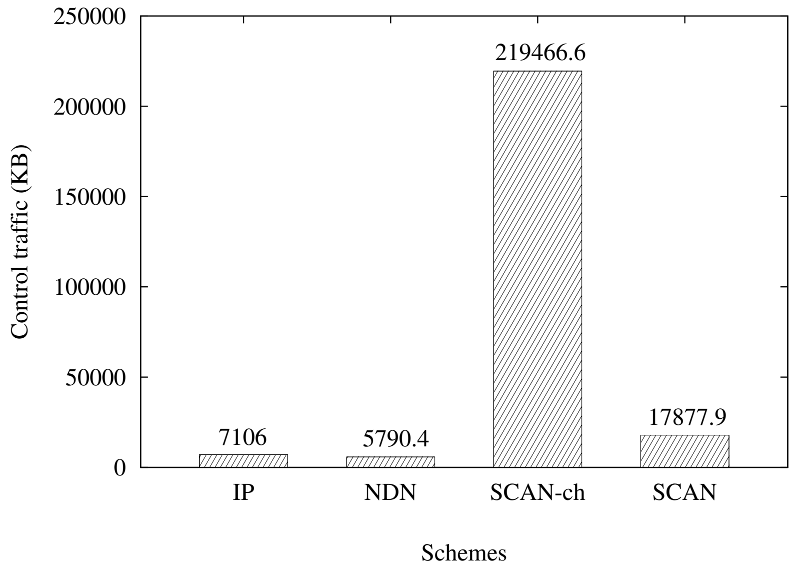


Fig. 2.6. Control traffic overhead.

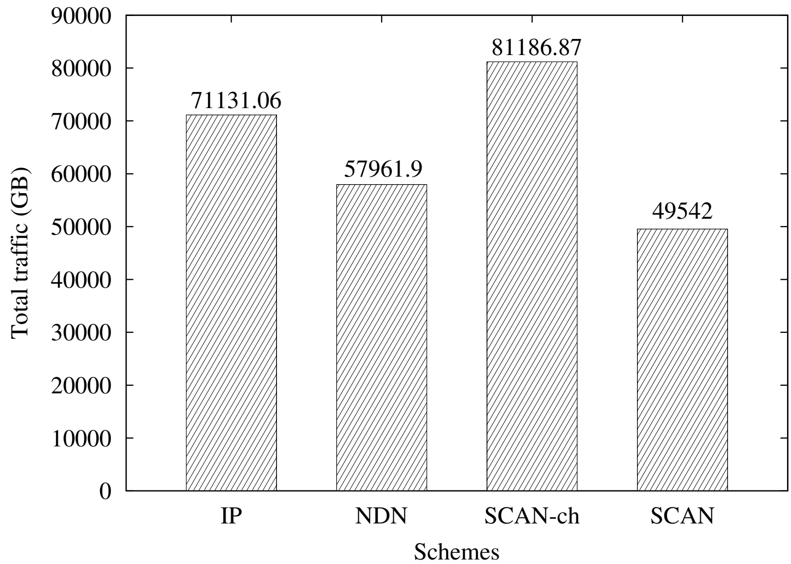


Fig. 2.7. Total traffic volume.

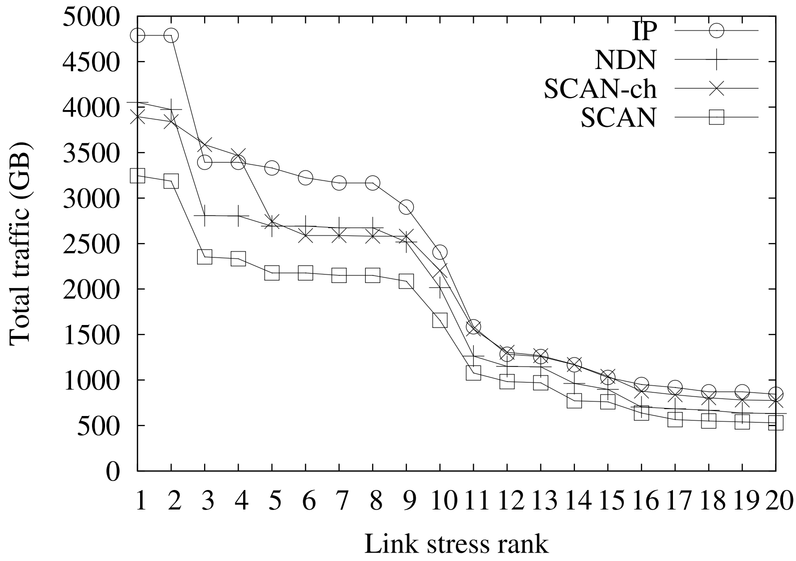


Fig. 2.8. Load balancing among links.

link stress, which is defined as the amount of traffic volume passed over a particular link. The top 20 stressed links are plotted in Fig. 2.8 in a descending order. These heavy-loaded links consist of inter-stub links and the links between transit routers and their first hop stub routers. Simulation result shows that SCAN achieves the lowest stress performance by utilizing closer cached contents. In addition, SCAN-ch shows higher stress performance than NDN since scanning operations for every chunk generates considerable redundant traffic.

2.5.3 Effect of Cache Size

To show the effect of cache size, we measure the inter-AS traffic and the total network traffic under different cache sizes, i.e., each C-router can store 2-12 GB

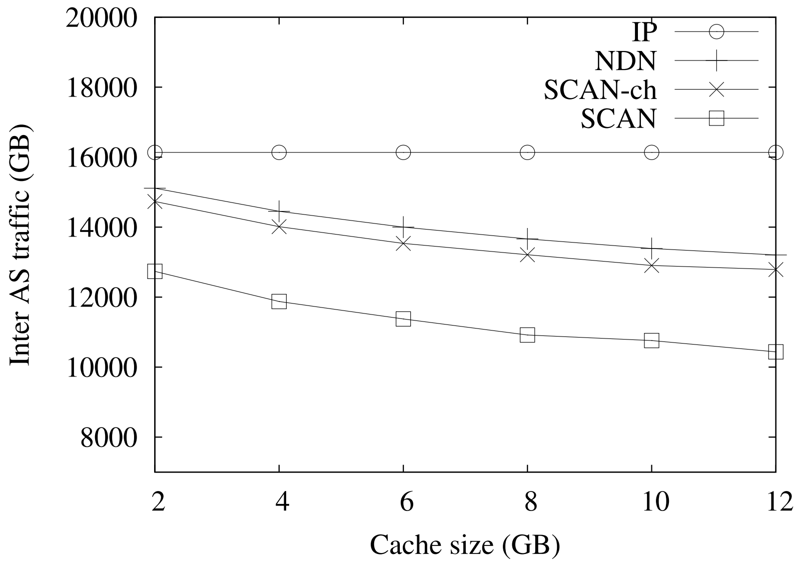


Fig. 2.9. The impact of cache size on Inter-AS traffic reduction.

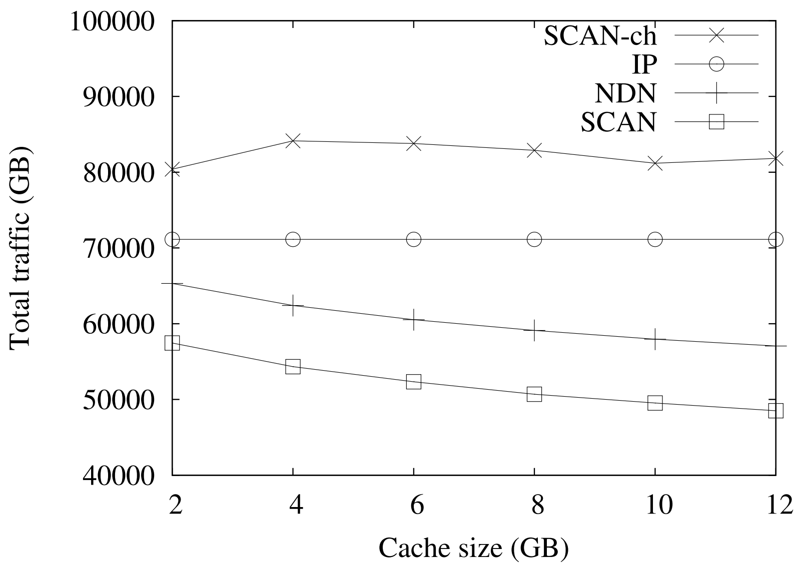


Fig. 2.10. The impact of cache size on Network traffic reduction.

of contents. Fig. 2.9 shows that schemes using in-network storage reduce inter-AS traffic as the cache size increases. This is because more contents are stored at large in-network storage during the content delivery and can be utilized for following requests. In particular, SCAN achieves the minimum inter-AS traffic since it exploits not only the en-route path, but also neighbor C-routers to find nearby cached contents. The inter-AS traffic of SCAN-ch is higher than that of SCAN due to the redundant traffic generated from scanning every chunk.

The volume of traffic reduction increases as the cache size increases, and SCAN and NDN significantly reduce the total traffic as compared to the IP routing described in Fig. 2.10. In Fig. 2.10, SCAN exhibits the lowest traffic volume. However, SCAN-ch has the highest total traffic volume, which fluctuates slightly with the cache size change. This can be explained as follows. In SCAN-ch, many scanning requests are generated as the number of cached contents increases. Thus, they result not only in redundant network traffic, but also frequent changes in cache status.

2.5.4 Effect of Scanning Depth

To show the efficiency of the scanning operation, we measure the average hop reduction and the traffic overhead while varying the scanning depth. As described in Chapter 2.3.3, the scanning depth is based on $min(\beta \times remainingHops, maxDepth)$. Thus, we change the value of $maxDepth$ from 1 to 3, which represents the scanning operation performed within 3-hop neighbors. We compare SCAN with NDN, SCAN-ch, and blind search (denoted by BlindSearch). Blind search does not maintain any cached content information and thus blindly floods the scanning requests from 1 to 3-hop neighbors. To investigate the effect of scanning depth in a controlled

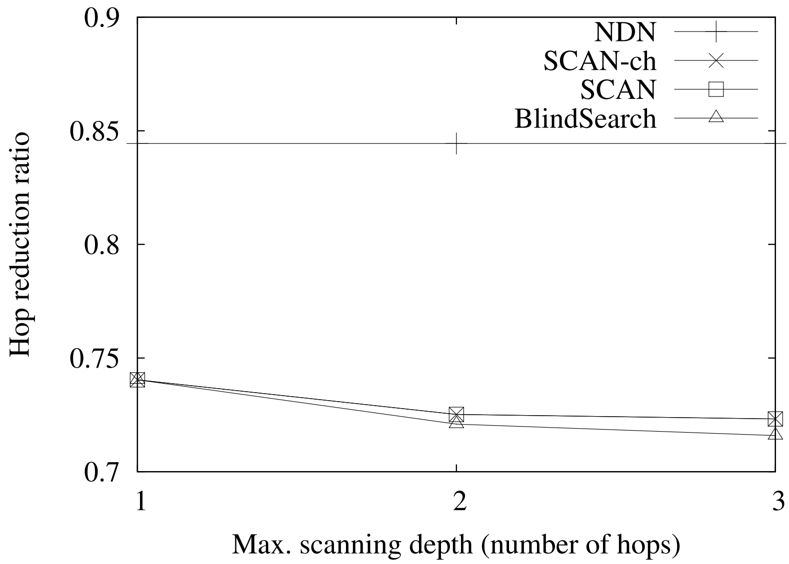


Fig. 2.11. The impact of scanning depth on Hop reduction.

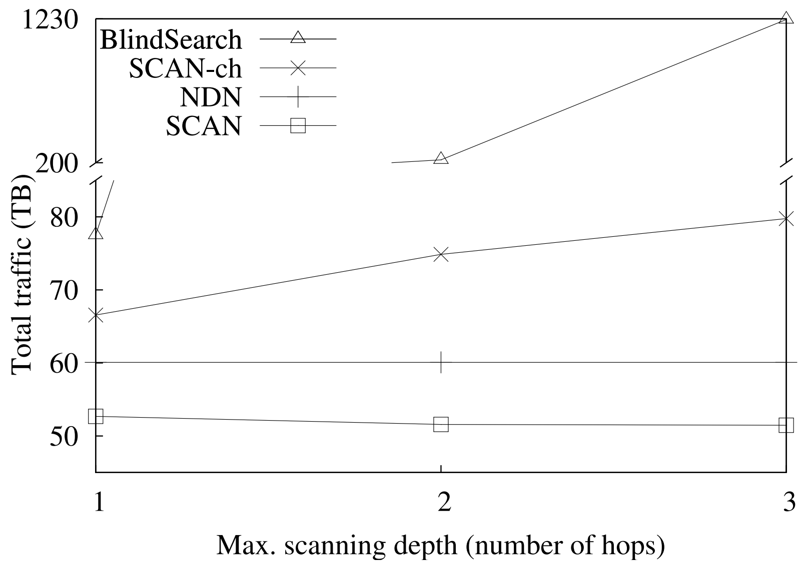


Fig. 2.12. The impact of scanning depth on Network traffic.

environment, we assume there is no cache replacement and thus cached contents are randomly distributed among C-routers following the Zipf distribution of content requests.

As shown in Fig. 2.11, SCAN, SCAN-ch, and blind search achieve about 2% reduction when *maxDepth* is changed from 1 to 2. Moreover, the hop reduction is insignificant when *maxDepth* is increased from 2 to 3. Fig. 2.12 shows the volume of network traffic in each scheme. As *maxDepth* (i.e., the scanning depth) varies from 1 to 3, there is no significant difference in the traffic volume of SCAN, but the traffic volume of SCAN-ch and blind search increase drastically. This is because SCAN-ch incurs a large redundant traffic overhead by sending so many scanning requests in a broad scanning range. Similarly, BlindSearch floods the scanning request blindly and thus generates a significant amount of network traffic. To conclude, these results reveal that SCAN performs the scanning operation more thoroughly while reducing unnecessary traffic.

2.5.5 Effect of Information Decay Probability

In SCAN, information decay is used to prevent all bits of BF from being set to 1 as the information of multiple hops is accumulated. Thus, there is a possibility that some content information that is cached over 2-hop away can be lost during CIB exchange. To see the effect of information decay, we vary the decay probability from 0.1 to 0.9. As shown in Fig. 2.13, the hop counts of SCAN and SCAN-ch slightly increase when the decay probability has a higher value. This is because the higher decay probability erases the information of cached content located over 2-hop away more aggressively. In contrast, NDN and BlindSearch send requests regardless of

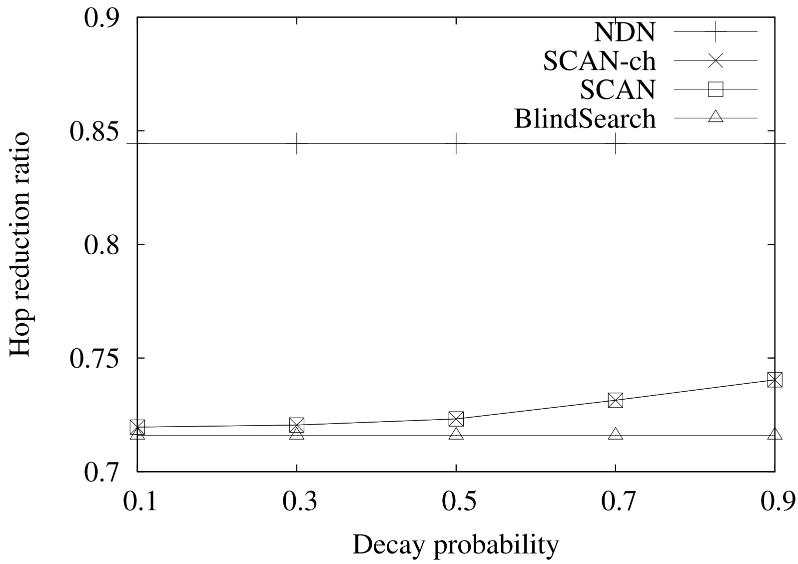


Fig. 2.13. The impact of information decay on hop reduction.

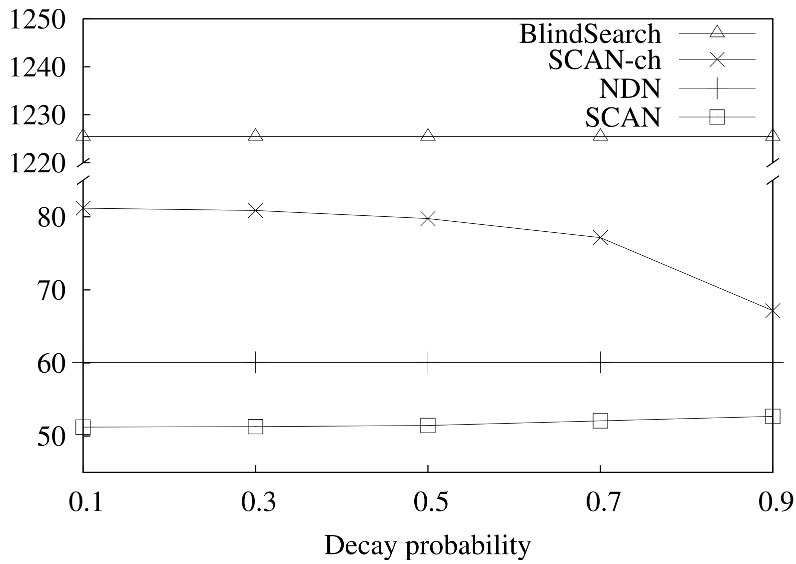


Fig. 2.14. The impact of information decay on network traffic.

CIB information. Thus, the hop count is not influenced by the decay probability.

Fig. 2.14 shows that SCAN achieves slightly increased total traffic volume as the decay probability increases. This is because SCAN occasionally retrieves content from a faraway C-router due to missing information, which increases network traffic. On the other hand, SCAN-ch benefits from traffic volume reduction as the decay probability becomes higher, because the lack of information generates fewer scanning requests in SCAN-ch.

2.5.6 Effect of BF Size

As the BF size decreases, the probability of false positives increases. Thus, more scanning requests can be propagated to irrelevant C-routers. As a result, unnecessary control traffic also increases. To show the effect of BF size on the probability of false positives, we change the size of the BF in CIBs from 100 to 1000 bits. As shown in Fig. 2.15, simulation results show that there are marginal differences in the overall performance of SCAN. In particular, the total traffic volume of SCAN is not affected by the BF size while the traffic of control overhead somewhat increases with a decrease in BF size. This desirable SCAN feature is from the partial content discovery and fallback mechanism that mitigate the impact of a false positive. When false positive occurs, an unnecessary scanning operation is performed since C-routers misjudge the information of the BF. However, extra scanning overhead only wastes small network traffic, and soliciting hosts can still utilize existing cached content. Even if there is no cached content in nearby C-routers, the fallback mechanism allows content retrieval from the publisher. Therefore, SCAN is able to mitigate the negative impact of a false positive.

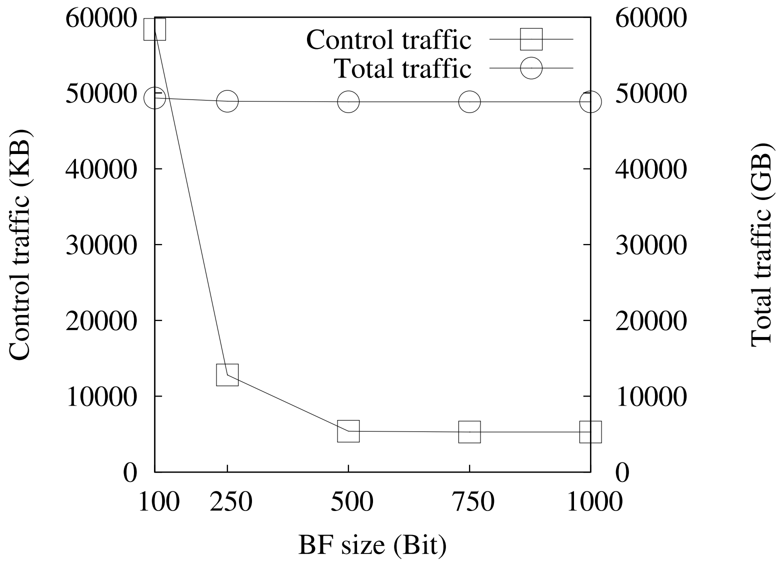


Fig. 2.15. The impact of the BF size on the false positive probability.

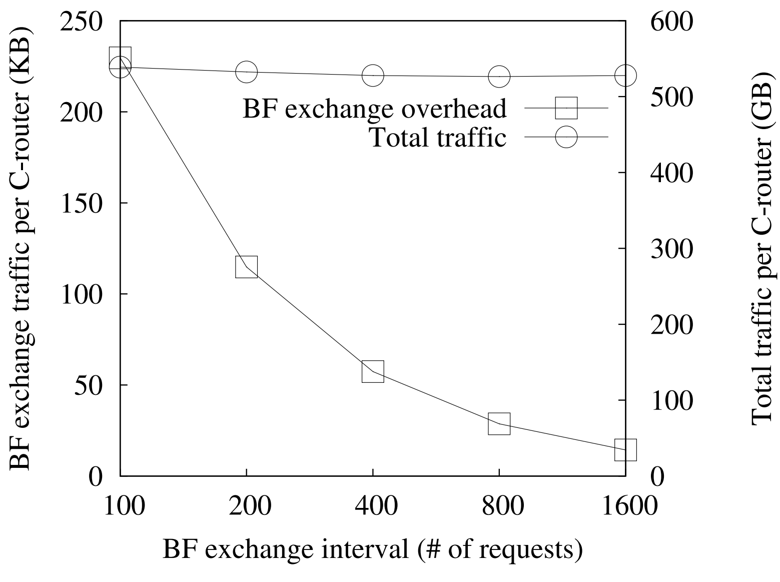
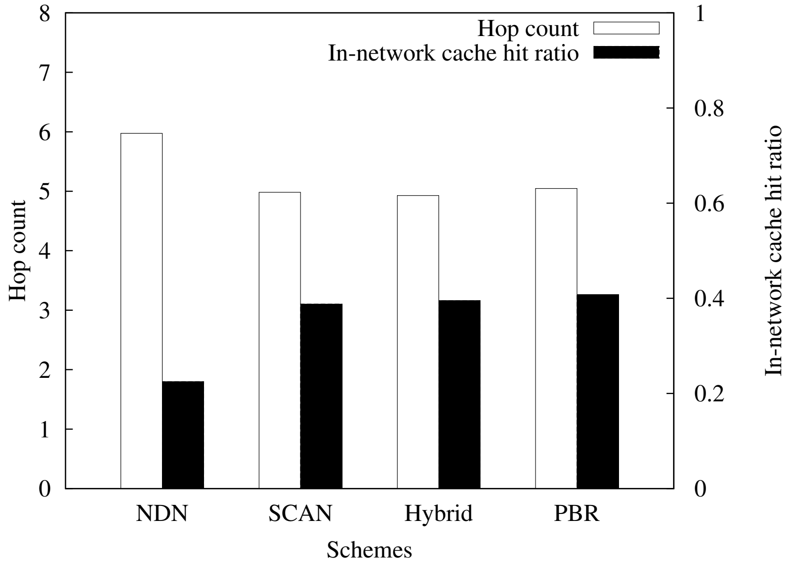
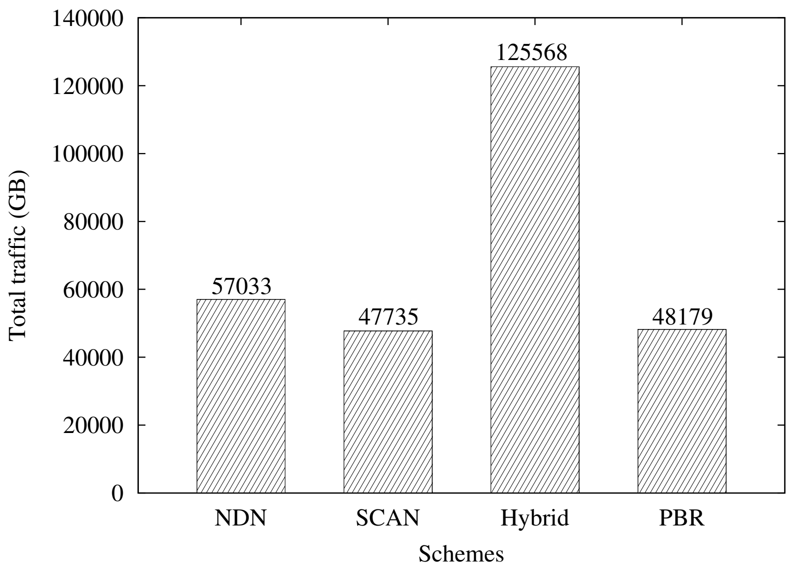


Fig. 2.16. Frequency of BF exchanges and traffic overhead.



(a) Comparison of Hop count and In-network cache hit ratio.



(b) Comparison of Total traffic.

Fig. 2.17. Performance comparison of ICN-enhancements.

2.5.7 Effect of BF Exchange Interval

In SCAN, the information of CIBs is updated by periodically exchanging compressed BFs among the neighbor routers. As the exchange interval is smaller, the BFs are exchanged more frequently. To show the effect of BF exchange interval, we measure the traffic overhead of BF exchange and the total network traffic volume, so the BF exchange interval is changed from 100 to 1600 requests. As shown in Fig. 2.16, C-router's traffic overhead for BF exchange decreases as the BF exchange interval increases. However, the difference of the total traffic volume is marginal, since the traffic overhead caused by exchanging compressed BF is about six orders of magnitude smaller than the total traffic volume.

2.5.8 Comparison with ICN-enhancements

In this chapter, we compare SCAN with two ICN-enhancements that are proposed in [20] and [21]. To locate cached contents, PBR [21] utilizes potential fields that maintain copies of content, while a hybrid forwarding mechanism [20] (denoted by Hybrid) combines a flooding mechanism and shortest path forwarding based on the content popularity. For comparison, each scheme is evaluated in the same simulation environments. We use the hop count, in-network cache hit ratio, and total traffic volume as performance metrics. The in-network cache hit ratio refers to how many portions of whole delivered content which comes from the cache all over the network, whereas the cache hit ratio is usually calculated on a single router. The cutoff parameter of Hybrid that decides the forwarding mechanism (flooding or shortest path) is set to 0.9 [20]. This value means the top 10% of popular contents are discovered by using the flooding mechanism.

Table. 2.2. Information Exchange Overhead per C-router

	Advertisement information	1-hop advertisement	2-hop advertisement	3-hop advertisement
PBR	1,821 bits[8]	1,593 Bytes	11,156 Bytes	78,096 Bytes
SCAN	2,500 bits	2,187 Bytes	2,187 Bytes	2,187 Bytes

Fig. 2.17(a) shows the average hop count and in-network cache hit ratio of each scheme. Compared to NDN, SCAN and ICN-enhancements show similar performance result. This is because these schemes discover off-path content replicas and retrieve the copy of content from a nearby cache. Fig. 2.17(b) shows the total network traffic volume. As shown in Fig. 2.17(b), Hybrid generates the highest traffic volume since it utilizes the flooding mechanism. Compared to NDN, SCAN and PBR achieve about 16% of traffic reduction because they retrieve a close copy of content. The traffic volume of SCAN shows somewhat lower than that of PBR since the content retrieval path of PBR is stretched when a request follows the potential field of removed cached content.

Even though the traffic volume of SCAN shows comparable results with that of PBR, the information exchange overhead of SCAN is smaller than that of PBR. Table. 2.2 shows the information exchange overhead per C-router according to the advertisement range. In our simulation, the average node degree of C-router is 7 and the cache size is 10 GB. Then, the size of advertisement information of C-router in SCAN is 2,500 bits (BF size) while each C-router in PBR advertises 1,821 bits of information. As shown in Table. 2.2, the information exchange overhead of PBR increases with the number of contents in the cache and the advertisement range, since a C-router in PBR advertises the potential fields of contents in its own cache, as well as n-hop neighbors' caches (i.e., 1-hop: 1,593 Bytes, 2-hop: 11,156 Bytes, 3-hop:

78,096 Bytes). On the other hand, the amount of information exchange overhead of SCAN remains constant (i.e., 2,187 Bytes) along with cache size or advertisement range, since SCAN compresses cached content information using BF and exchanges aggregated BFs. In summary, ICN-enhancement schemes including SCAN can enhance the original NDN by utilizing nearby cached content efficiently. However, the design principle of SCAN makes the difference in terms of traffic volume.

Chapter 3

Data Offloading for Information-Centric Vehicular Networks

3.1 Introduction

In wireless environments, the increasing mobile traffic is becoming a serious concern for mobile network providers. if in-vehicle data services using cellular networks become prevalent, it will significantly worsen the problem of cellular traffic explosion [18]. To address this traffic explosion problem not only in vehicular environments, but also in general settings, there have been a lot of efforts to offload the traffic from cellular networks to other networks, such as WiFi hotspots and femtocells [36–38]. Also, recent measurement study [39] shows that the requests of some popular videos account for the majority of all the requests, and a significant amount of cellular traffic is redundant. Thus, we focus on the data offloading for redundant traffic caused by in-vehicle data services in order to mitigate the traffic overload in cellular networks. To this end, a *natural research question* is how to design an efficient offloading framework utilizing components of vehicular networks (i.e., relay nodes) to provide the delay-insensitive in-vehicle data services, while minimizing the usage of expensive cellular links.

To design an offloading framework leveraging vehicular networks, we focus on the following characteristics of vehicular networks. First, a vehicle is equipped with multiple communication interfaces (e.g., DSRC and 4G-LTE), a GPS navigation system, and a storage module. Thus, vehicles can carry volume-intensive files like multimedia on their storage and communicate with each other by DSRC communications. Second, the vehicle mobility is predictable since (i) vehicles are moving along the constrained roadways and (ii) their travel paths (i.e., navigation paths) to destinations can be calculated from navigation systems. Last, infrastructure nodes (e.g., RNs [15, 16]) that are deployed in vehicular networks for the driving safety [40] can be used for data offloading. With the mobility information and infrastructure nodes of vehicular networks that are available for data offloading, a significant amount of the cellular traffic for vehicles can be reduced.

In this chapter, we propose a Data Offloading framework through Vehicular nEtworks (DOVE) that can significantly reduce the cellular traffic for delay-insensitive in-vehicle data services in a cost-effective way. The key idea in DOVE is to predict vehicle mobility with vehicle trajectories provided by the GPS navigation systems. With this vehicle mobility prediction, DOVE selects appropriate *offloading positions (OPs)* where vehicles can retrieve the requested files without using cellular links. Note that RNs play the role of OPs since RNs can keep and deliver files to the passing vehicles that request the files. In this work, we formulate the selection of OPs (i.e., RNs) as a spatio-temporal set-covering problem where each offloading position covers a set of vehicles that request the same file. To solve the selection of OPs, we propose a *time-prediction based set-covering algorithm* (called *DOVE algorithm*) using vehicle trajectories. This *DOVE algorithm* selects approximately the minimum number of OPs, reducing a significant amount of aggregated usage of cellular links for vehicles. Our contributions in this work are as follows:

- **DOVE**: we propose a data offloading framework (called *DOVE*) using the components of vehicular networks (i.e., RNs) for delay-insensitive in-vehicle data services.

- **DOVE algorithm**: we formulate the selection of offloading positions as a spatio-temporal set-covering problem. To solve this problem, we propose a time-prediction based set-covering algorithm (called *DOVE algorithm*) using vehicle trajectories to select offloading positions.

The rest of this chapter is organized as follows. Chapter 3.2 summarizes related work. Chapter 3.3 describes a problem formulation. Chapter 3.4 explains the design and operations of DOVE framework. In Chapter 3.5, we evaluate the performance of our DOVE algorithm.

3.2 Related Work

In vehicular networks, research about data forwarding and data dissemination has been investigated. *TBD* [41], *TSF* [15], *GeOpps* [42], and Leontiadis *et al.* [43] utilize vehicle trajectories along with vehicular traffic statistics to forward packets with shorter delivery delay and better delivery probability. For the data dissemination, Leontiadis *et al.* [44] propose a content based information dissemination protocol. Their protocol utilizes vehicle trajectories in order to disseminate some information to specific areas where vehicles need to receive it. Similarly, We *et al.* [45] propose *MDDV* that exploits a predefined trajectory for data dissemination. For all those existing approaches, authors focus on vehicle trajectories to forward/disseminate an information to vehicles in the specific location. On the other hand, DOVE investigates how to utilize both vehicle trajectories and potential benefits of the vehicular networks for offloading purposes.

There have been several studies to offload cellular traffic using WiFi or opportunistic communications [36–38, 46, 47]. Lee *et al.* [36] show that the offloading of 3G traffic to WiFi can significantly benefit mobile providers in terms of infrastructure cost. In *BreadCrumbs* [37], authors show that the forecast of WiFi access can improve the offloading of several applications. Also, recent studies [46–48] exploit mobile devices of social friends to offload cellular traffic through opportunistic communications. These previous studies focus on the data offloading using WiFi hotspots and client devices. In contrast, DOVE investigates the utilization of vehicular infrastructure (i.e, relay nodes) for data offloading.

Several works [18, 49, 50] address the mobile data offloading problem in vehicular networks. Li *et al.* [18] conduct the mathematical analysis with optimization problem of the offloading in vehicular networks. Also, Malandrino *et al.* [50] investigate a content prefetching at road-side units (RSUs) via the Internet. In [49], Siris and Kalyvas propose an offloading scheme that exploits WiFi hotspots located in single vehicle's route. For data offloading, above works focus on WiFi or RSUs, which are deployed with the Internet connectivity. Our work differs in that we utilize relay nodes, which do not have the Internet connectivity for cheaper installation cost, by considering vehicle trajectories of multiple vehicles to offload the redundant cellular traffic of vehicles.

3.3 Problem Formulation

3.3.1 Target Scenario and Goal

A significant amount of cellular traffic is redundant since multiple users repeatedly download the same popular file [39]. Accordingly, a substantial reduction of cellular traffic for vehicles is expected by offloading the duplicated traffic (of popular files) to vehicular networks. Thus, *our target* is to reduce the cellular traffic of popular files for delay-insensitive in-vehicle data services. Target content files can be (i) update files for software in car system and (ii) popular multimedia files (e.g., headline news, music files, and YouTube video clips).

If these content files are delay-insensitive, it can be assumed that users are willing to wait for some delay to reduce the cost of cellular links [38]. In other words, users in vehicles face a trade-off between cost and delay in making their offloading decisions. In our scenario, vehicles try to offload their traffic from cellular links to RNs in vehicular networks when users choose to wait in order to reduce their cellular service cost.

In our target scenario, *our goal* is to select effective offloading positions (OPs), such as RNs that minimize the aggregated usage of cellular links for vehicles (i.e., the amount of data downloaded through cellular networks), while satisfying the user-defined *quality of experience (QoE)* requirements in terms of content retrieval delay.

3.3.2 DOVE Components and Assumptions

As depicted in Fig. 3.1, we describe the components of vehicular offloading framework (DOVE) with assumptions.

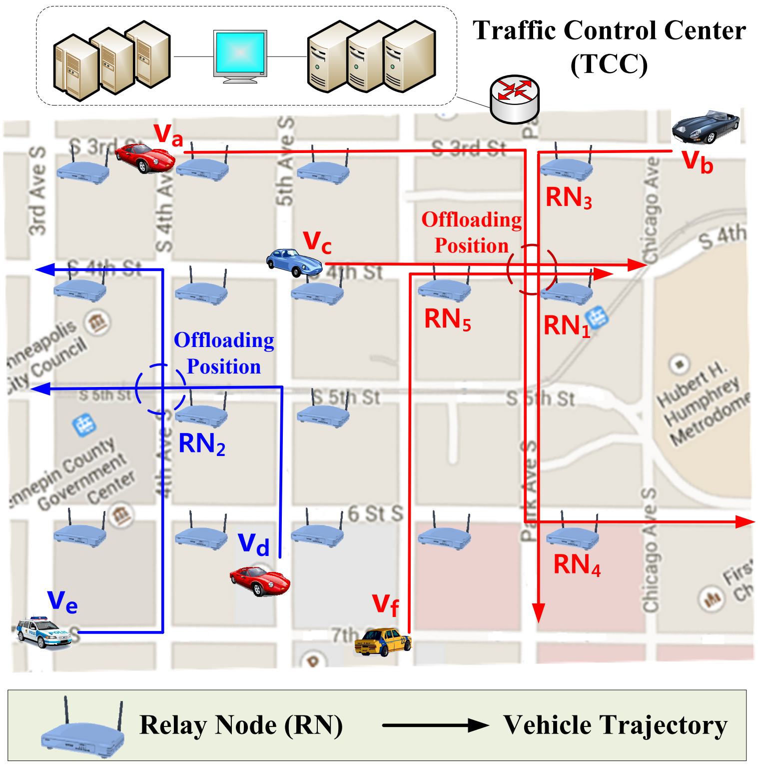


Fig. 3.1. Data offloading framework using RNs in vehicular networks.

◇ **A traffic control center (TCC)** [15, 51] is a traffic management node that maintains vehicle trajectories for the location management of vehicles. In DOVE, the TCC collects the requested content information from vehicles. Thus, it knows the set of vehicles (called *request vehicles*) that request the same content file. Note that vehicles can communicate with the TCC via cellular links, whose traffic volume is negligible. Also, multiple TCCs can be used for the scalability and survivability by dividing the road network into several regions.

◇ **A relay node (RN)** is a wireless packet holder for the reliable forwarding, which is usually deployed in vehicular networks for the driving safety [15,40]. Compared with a road-side unit (RSU) [50], an RN also has the DSRC communications and storage modules, but does not have the Internet connectivity for the cost effectiveness. Note that the cost for installing an RSU is as much expensive as US $5,000 [52] due to the installation of wireline for Internet connectivity. Since RNs can keep and share content files using their storage, we can exploit RNs for data offloading (role of OPs) as well as driving safety. It is assumed that one RN is deployed at each intersection for the driving safety [40]. However, DOVE works even in the case where some intersections do not have their own RNs. The details will be described in Chapter 3.5.5.

◇ **Vehicles** participating in vehicular networks have DSRC devices and cellular communication devices [14, 17]. Thus, vehicles can communicate with RNs using DSRC devices to retrieve the requested file.

◇ Vehicles, the TCC, and the RNs are equipped with GPS navigators and digital road maps. Recent commercial navigation systems provide vehicles with the traffic statistics information, such as average vehicle speed v and vehicle arrival rate λ per road segment [53].

◇ When users in vehicles want to retrieve a content file, they should make a decision: (i) consuming cellular links, immediately, with cellular link cost, or (ii) waiting for offloading via vehicular networks, within tolerance time δ, without cellular link cost. If users decide to use delay-insensitive content retrieval through data offloading, they send their current location, trajectory, and content request to the TCC via cellular links. Note that TCC can perform the data offloading schedule with the vehicle trajectories. Even though all of the drivers are not using GPS navigators for their everyday commute, we assume that the drivers participating in DOVE service are using GPS navigators. For the vehicles without GPS navigator, we do not count them as DOVE users.

3.3.3 Design Principles using RNs

DOVE is designed to use RNs for the role of OPs due to the following reasons: (i) *High success probability* and (ii) *Cost effectiveness*. First, to use vehicles as OPs, vehicles requesting the same file should encounter each other and establish connectivity within the DSRC range. However, the probability of such occasions will be very small. On the other hand, when we use static infrastructure nodes (i.e., RNs) as OPs, the file can be shared with any vehicles that will pass the RNs. The vehicles reaching RNs with the file can get data from the RNs at any time. Thus, the data offloading using these RNs can provide reliable data delivery for moving vehicles. Second, the cost for installing an RN is relatively cheaper than that of an RSU [15, 52]. The reason is that as stand-alone wireless nodes, RNs do not have the Internet connectivity unlike RSUs.

3.3.4 The Concept of Offloading in DOVE

RNs are infrastructure nodes usually deployed at intersections for the driving safety [40]. In DOVE, RNs play the role of OPs for offloading purposes. Given the request vehicles and their trajectories, we can find RNs where trajectories are overlapped. We call them the candidates for OPs and we will select appropriate RNs among the candidates. In Fig. 3.1, RN_1-RN_5 are candidates for OPs. If a selected RN has the requested file, vehicles passing through the selected RN can retrieve the file by DSRC communications. Since the content file is retrieved from the vehicular network, vehicles can reduce the cost of cellular link usage. In this case, the selected RN serves as a *spatio-temporal rendezvous* for request vehicles. Thus, vehicles can offload their traffic from the cellular links to the DSRC links to RNs in the vehicular network.

To use the selected RN as an OP, the RN should hold the file for *request vehicles* passing through the RN. To make the requested file be located in the selected RN, a vehicle that will arrive first at the selected RN is scheduled to download the file using cellular links. Also, it will store the downloaded file into the selected RN for other *request vehicles*. We call this vehicle a *provider*. On the other hand, other *request vehicles* except the *provider* are scheduled to retrieve the requested file from the selected RN without using cellular links. We call these vehicles *consumers*. In Fig. 3.1, RN_1 is selected as the OP and vehicle v_c is a provider. Thus, the provider v_c stores the file in RN_1 and consumers v_a, v_b, and v_f can retrieve it via the vehicular network. In summary, when an RN is selected as an OP, the RN receives the content file from the provider, keeping the file in its local storage. The stored content file can then be delivered by the RN to other vehicles (i.e., consumers) when they reach the communication range of the RN.

3.4 Design and Operations of DOVE

In this chapter, we explain the design and operations of our DOVE framework. First, we model the travel time of vehicle. Second, we explain the design and operations of our DOVE through a selection algorithm for offloading positions along with the selection of providers as follows: (i) the collection of content requests of vehicles by TCC, (ii) the selection of OPs, (iii) the selection of providers, and (iv) the selection of comsumers.

3.4.1 Travel Time Prediction

We model the travel time of a vehicle from one position to another position in a given road network. Using the travel time prediction and vehicle trajectories, the arrival time of vehicle at a particular RN can be calculated.

3.4.1.1 Travel Time through Road Segment

The travel time of vehicle over a fixed distance follows the Gamma distribution in light-traffic road condition [15, 54]. Thus, the travel time through a road segment i in the road network (called *link travel time*) is modeled as: $d_i \sim \Gamma(\kappa_i, \theta_i)$ where κ_i is a shape parameter and θ_i is a scale parameter. The parameters κ_i and θ_i are computed with the mean travel time μ_i and the travel time variance σ_i^2 [15]. Note that the traffic statistics of μ_i and σ_i^2 can be computed by commercial navigation service provider [53].

Let the mean of d_i be $E[d_i] = \mu_i$ and the variance of d_i be $Var[d_i] = \sigma_i^2$, the formulas for κ_i and θ_i are as follows:

$$\theta_i = \frac{Var[d_i]}{E[d_i]} = \frac{\sigma_i^2}{\mu_i} \tag{3.1}$$

$$\kappa_i = \frac{E[d_i]}{\theta_i} = \frac{\mu_i^2}{\sigma_i^2} \tag{3.2}$$

3.4.1.2 Travel Time on End-to-End Path

As described above, the travel time through a road segment i is modeled as the Gamma distribution of $d_i \sim \Gamma(\kappa_i, \theta_i)$. Given a vehicle trajectory, we assume that the travel times of road segments consisting of the trajectory are independent. Under this assumption, we approximate the mean and variance of End-to-End (E2E) travel delay as the sum of the means and the sum of the variances of the link travel times along the trajectory, respectively. Assuming that the traveling path consists of N road segments, the mean and variance of the E2E travel delay D are computed as follows:

$$E[D] = \sum_{i=1}^{N} E[d_i] = \sum_{i=1}^{N} \mu_i \tag{3.3}$$

$$Var[D] = \sum_{i=1}^{N} Var[d_i] = \sum_{i=1}^{N} \sigma_i^2 \tag{3.4}$$

With (3.3) and (3.4), the E2E vehicle delay distribution can be modeled as a Gamma distribution as follows: $D \sim \Gamma(\kappa_D, \theta_D)$ such that κ_D and θ_D are calculated using E[D] and Var[D] using the formulas of (3.1) and (3.2). Note that we can use any better E2E travel time distribution if it is available from actual measurement or mathematical model.

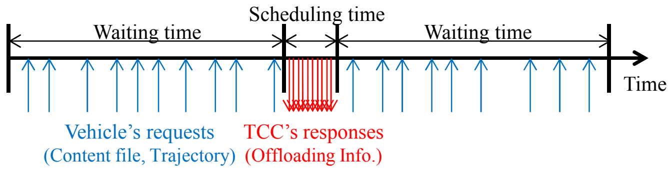


Fig. 3.2. The operation of the TCC.

Using the travel time prediction, we can estimate vehicle's arrival time at a particular RN as follows: Let T^* be the current time. Let $t_{a,b}$ be the E2E travel time from the current position a to the RN b. We define $T_{v,p_v i}$ as the vehicle v's arrival time at RN i from the current position p_v. Then, the arrival time can be modeled as a Gamma distribution with Equations (3.3) and (3.4) such that $T_{v,p_v i} = T^* + t_{p_v,i}$.

3.4.2 The Operation of TCC

To decide which RNs are used as OPs, the TCC collects the content request and the trajectory from each vehicle. As shown in Fig. 3.2, the TCC periodically performs a scheduling operation. In the scheduling operation, the TCC constructs a set of *request vehicles* with vehicles requesting the same file. Also, it finds the candidates for OPs as RNs through which the trajectories of the vehicles pass. Next, the TCC tries to select the optimal OPs among the OP candidates and the providers for the selected OPs by the *selection algorithm*, described in Chapter 3.4.3. Finally, the TCC sends the offloading information to each vehicle. The offloading information includes the information of an OP and the role of a vehicle (i.e., *provider* or *consumer*). If there is no available OP, vehicles receive this information from the TCC and use cellular links to download the file.

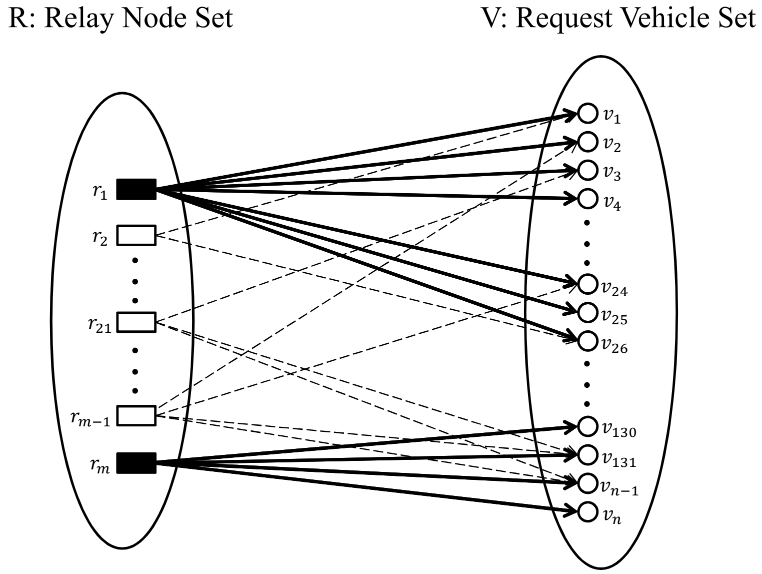


Fig. 3.3. The set-cover of RN set for request vehicle set.

3.4.3 The Selection Algorithm for Offloading Positions

In this chapter, we formulate the selection of OPs as a spatio-temporal set-covering problem, and propose a time-prediction based set-covering algorithm (called *DOVE algorithm*). Selecting an effective OP is important because the reduced cellular traffic volume and the content retrieval delay are determined by the OP. To reduce the aggregated usage of cellular links for vehicles, the number of providers using cellular links should be minimized. Since each provider consumes cellular traffic to provide the requested content file with an appropriate OP, the reduction of OPs can lower the number of providers. Therefore, we try to minimize the aggregated cellular traffic by selecting the minimum number of OPs. In other words, *the problem of minimizing cellular traffic can be formulated as the problem of selecting the minimum number of*

OPs.

Let R be a set of RNs in the road network and V be a set of request vehicles requesting the same content file. Then, the selection of OPs can be formulated as the relation between set R and set V. In Fig. 3.3, a line (both solid and dashed) between an RN and a vehicle means that the vehicle will traverse the RN and hence can retrieve the file from the RN. If we select r_1 and r_m as OPs, they can *cover* all the request vehicles (represented by solid lines). This relation can be formulated as a *set-covering problem* that is one of NP-hard problems.

Let V be the set of request vehicles, and let S_i be the set of request vehicles covered by an RN i. Let F be a family of subsets of V, that is, $S_i \subseteq V$ such that $V = \bigcup_{S_i \in F} S_i$ where $F = \{S_i | i \in R\}$. Also, we define the collection of subsets $C \subseteq F$ as the set-cover of V such that $V = \bigcup_{S_i \in C} S_i$. Then, our set-covering problem is to find a minimum set-cover C^* of RNs as OPs as follows:

$$C^* \leftarrow \arg\min_{C \subseteq F} |C|, \tag{3.5}$$

where $V = \bigcup_{S_i \in C} S_i$. Let the set P be the set of offloading pairs, which consist of an RN and a provider. The identifier i of the element sets $S_i \in C$ means the selected RN i as an OP, which covers the request vehicles in S_i. Next, we propose a time-prediction based set-covering algorithm (called *DOVE algorithm*). Also, we propose an enhanced version of *DOVE algorithm* to further reduce the usage of cellular links.

3.4.3.1 DOVE Algorithm

We design the DOVE algorithm based on the greedy approach because it is known as the best possible polynomial time approximation algorithm for the set-covering algorithm under reasonable complexity assumptions [55]. Given a set of RNs R and a set of request vehicles V, the idea is to select the set S_i covering the largest number of the remaining vehicles not covered yet in each step in order to select the next RN as an OP, considering the travel times of vehicles.

Algorithm 3.1 DOVE Algorithm (R, V, F)

$I \leftarrow R$
$U \leftarrow V$
$P \leftarrow \emptyset$
while $U \neq \emptyset$ **do**
    update $S_i^* \leftarrow S_i$ for $i \in I$ by **pruning** unsatisfied vehicles v
    such that $\hat{t}_{p_v,i} < \gamma$ or $t_{p_v,i} > \delta$ where $v \in S_i$.
    select a $S_i^* \in F$ that maximizes $|S_i^* \cap U|$ for $i \in I$
    select a provider $d_i \in S_i^*$ whose arrival time at RN i is minimum
    $U \leftarrow U - S_i^*$
    $I \leftarrow I - \{i\}$
    $P \leftarrow P \cup \{(i, d_i)\}$
end while
return P

Let $\hat{t}_{p_v,i}$ be the travel time of vehicle v from the current position p_v to out of the communication range of an OP i; note that in this case, the vehicle v is within the communication range of the OP i. Let $t_{p_v,i}$ be the travel time of v from the current position p_v to an OP i, as defined in Chapter 3.4.1.2. We use γ to denote the sum of cellular download time and DSRC upload time when a provider provides a file for an RN via cellular and DSRC links. The γ value is decided by the *size of file* and the *bandwidthes of cellular and DSRC links*. In Algorithm 3.1, the set S_i is updated considering vehicles' travel times. First, a request vehicle (provider or consumer) that

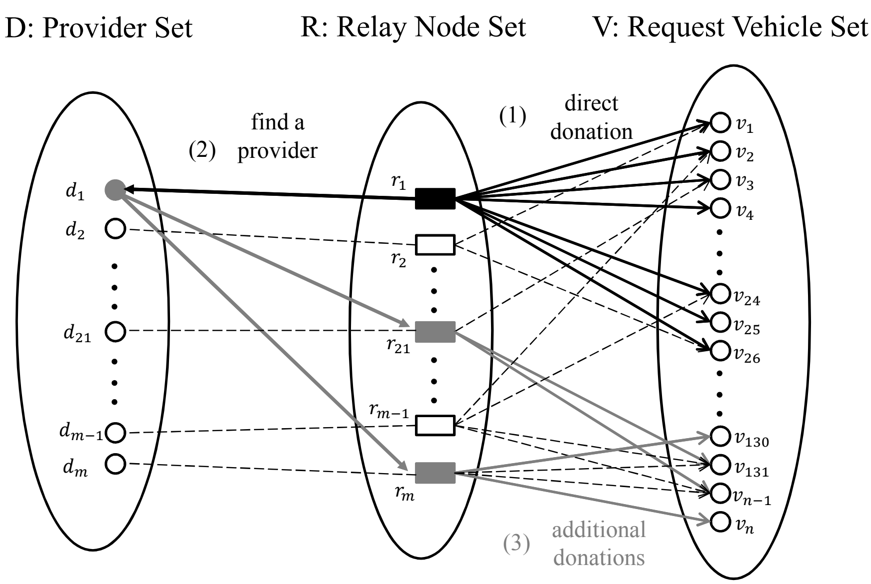


Fig. 3.4. The basic idea of multiple donations (DOVE$^+$).

cannot satisfy the condition $\hat{t}_{p_v,i} < \gamma$ is removed from the set S_i. In this case, a provider cannot finish downloading the file using cellular links until it leaves the communication range of the OP, or a consumer will leave the OP before the file is ready. Second, a consumer in the set S_i is excluded when its expected content retrieval delay is longer than the tolerance time constraint δ. After pruning unsatisfiable vehicles, the DOVE algorithm selects an RN that covers the largest number of vehicles. Then, we select the first arriving vehicle at the OP as a provider from the set S_i^*. Finally, we can obtain the approximately minimum number of OPs with providers.

3.4.3.2 DOVE$^+$ Algorithm (using Multiple Donations)

In the DOVE algorithm, each provider is scheduled to store the downloaded file only at a single OP for other consumers. However, providers can put the file at multiple OPs since they will pass multiple RNs. Thus, we can enhance the DOVE algorithm to take advantage of multiple donations to further reduce the number of providers that consume cellular traffic. Fig. 3.4 shows the basic idea using multiple donations (drawn by the gray lines). We call this enhanced algorithm **DOVE$^+$**. The following procedure of DOVE$^+$ algorithm is the same as the DOVE algorithm except step *(3) additional donations*:

(1) Select an S_i^* to decide an OP that directly covers the maximum vehicles (line 6 in Algorithm 3.1). In Fig. 3.4, RN r_1 is selected as an OP. Then, it can cover vehicles v_1-v_4 and v_{24}-v_{26}.

(2) Find a provider that is the first vehicle reaching the selected OP (line 7 in Algorithm 3.1). In Fig. 3.4, a vehicle d_1 is selected as a provider for RN r_1.

(3) Find additional consumers using RNs where a provider will pass (multiple donations). In Fig. 3.4, d_1 can store the file in RN r_{21} and r_m, so this can additionally cover vehicles v_{130}, v_{131}, v_{n-1}, and v_n.

(4) Repeat the steps (1)-(3).

Clearly, there is a trade-off between DOVE and DOVE$^+$ in terms of the reduced cellular traffic and the content retrieval delay. DOVE$^+$ algorithm reduces the aggregated usage of cellular links for vehicles while increasing the average content retrieval delay. We can select one depending on the requirements. Note that the time complexity of DOVE or DOVE$^+$ algorithms is computed as $O(VF \cdot min(V,F))$, which is a polynomial time [55].

3.4.4 The Selection of Providers

In this chapter, we describe how we select providers. When the requested file is located at the OP, the consumers passing through the OP can offload their cellular traffic to the OP by DSRC communications. To cover as many vehicles as possible, the TCC selects the first vehicle reaching the communication range of the OP as a provider. The arrival time of vehicles is calculated using the time prediction model, as described in Chapter 3.4.1. Note that incentives for providers can be rewarded by mobile providers (e.g., reduced cost for using their cellular networks). The incentive policy for data offloading is left as future work.

3.4.5 The Operation of Vehicles using Offloading Positions

When a vehicle receives the offloading information from the TCC, it can know its OP and whether it is a provider or not. If a vehicle is selected as a provider, the vehicle downloads the file over cellular link. Then, the provider stores the file in the scheduled OP over DSRC link. On the other hand, if a vehicle is decided as a consumer, it postpones downloading the file. Then, the vehicle retrieves the file from the scheduled OP over DSRC link. Since the file is retrieved by DSRC link, the consumer can reduce the cost of cellular link usage. Note that vehicles should consume cellular traffic in the following cases: (i) vehicles are notified as providers by the TCC, (ii) vehicles fail to receive the file from the scheduled OP, and (iii) vehicles do not retrieve the file within the tolerance time.

Table. 3.1. Simulation Configuration (DOVE)

Parameter	**Description**
Road network	The number of intersections is 49. The area of the road map is 8.25km$\times$9km.
Communication range of DSRC	Communication range $R = 200$ meters. Bandwidth of the DSRC = 25 Mbps [14].
Number of vehicles (N)	The number of vehicles moving within the road network. The default N is 300.
Vehicle speed (v)	$v \sim N(\mu_v, \sigma_v)$ where $\mu_v = \{20, 25, ..., 60\}$ MPH and $\sigma_v = 5$ MPH [15,54]. The maximum and minimum speeds of vehicles are $\mu_v + 3\sigma_v$ and $\mu_v - 3\sigma_v$, respectively. The default (μ_v, σ_v) is $(40, 5)$.
Deployment ratio (α)	The ratio α of the number of deployed RNs to the total number of intersections. The default α is 100%.
Tolerance time (δ)	The maximum (tolerable) delay of vehicles. The default δ is 600 sec (i.e., 10 min) [38].
Cellular downloading time (γ)	The content downloading time through a cellular network. Cellular downloading time γ is 47.9 sec. (Assumption: file size=12 MB, BW=2.1 Mbps)

3.5 Performance Evaluation

In this chapter, we evaluate the performance of DOVE. Since we have no other state-of-the-art algorithm for OP selection, we compare the selection algorithm of DOVE (called DOVE) with the following: (i) a random selection algorithm that randomly selects OPs (called *Random*), and (ii) a greedy selection algorithm that selects an OP that covers the largest number of request vehicles without time consideration (called *Greedy*), and (iii) DOVE$^+$ algorithm (called *DOVE*$^+$).

To this end, we have developed a packet-level discrete event simulator using the SMPL [56]. Table 3.1 summarizes the detailed description of the simulation configuration. We use a road network which consists of 49 intersections. The layout of the road network is based on the map of Minneapolis downtown in Minnesota in the

US. Each vehicle's movement pattern is determined by a Hybrid Mobility model of City Section Mobility model [57] and Manhattan Mobility model [58]. To reflect the stop sign or traffic signal, each vehicle waits for a random (uniform) waiting time between 0 and 10 sec at each intersection [15]. Each vehicle's speed is generated from a normal distribution of $N(\mu_v, \sigma_v)$ [15, 54, 59] as described in Table 3.1. The communication-related parameters (e.g., DSRC communication range) in our evaluation are selected, based on a typical DSRC scenario [14]. To share a file in our simulations, each request vehicle sends a content request to the TCC along with its trajectory and current location information. Note that 5% of vehicles are selected as request vehicles that request the file. The total size of the request vehicle set is 1,000 and we use the averaged evaluation results. Unless otherwise specified, the default values of parameters in Table 3.1 are used.

3.5.1 Overall Performance of Data Offloading

We compare the offloading performance of DOVE, *Greedy*, *Random*, and *DOVE*$^+$. Note that we assume a perfect time-prediction solution of DOVE for comparison purposes. The perfect time-prediction solution (called *Perfect*) is assumed that the TCC can estimate the travel times of vehicles without any error caused by traffic signals. Thus, all the vehicles perfectly perform offloading operations according to the TCC's schedule.

As shown in Fig. 3.5, we analyze the portions of *the number of vehicles using the cellular network* and *the number of vehicles using the vehicular network* to the total number of request vehicles, respectively. When OPs are selected by DOVE, more than a half of request vehicles (i.e., 57%) offload their traffic from the cellular

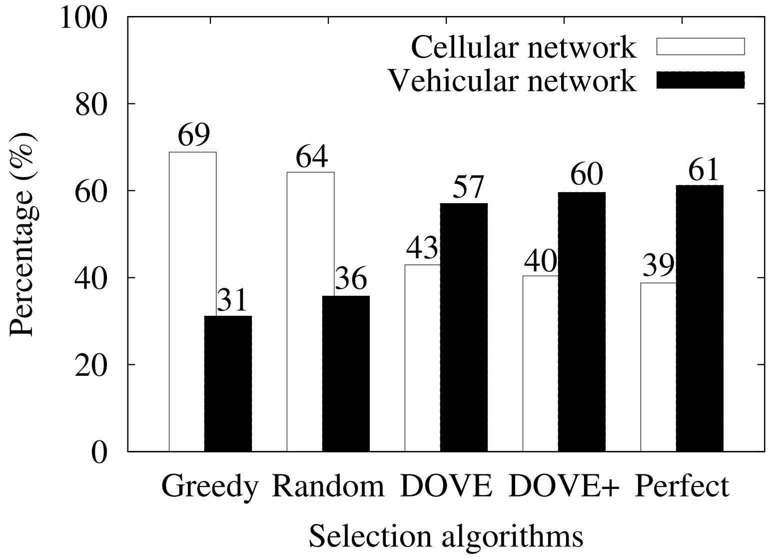


Fig. 3.5. The ratio of cellular network usage and vehicular network usage.

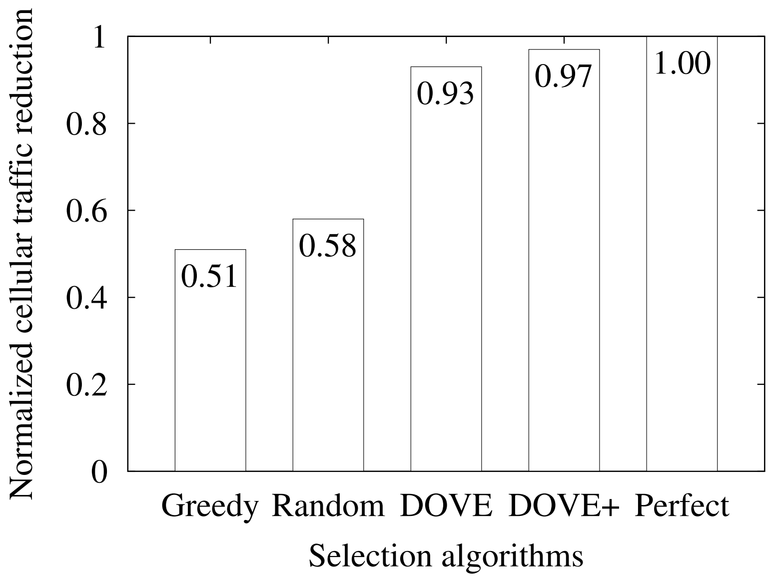


Fig. 3.6. The normalized cellular traffic reduction by offloading.

network to the vehicular network. Also, the offloading ratio of *DOVE*$^+$ is somewhat increased (i.e., 60%) since *DOVE*$^+$ reduces the number of providers by multiple donations. Compared to *Perfect* (61%), DOVE and *DOVE*$^+$ show almost comparable performance. However, only 31% and 36% of request vehicles take advantage of data offloading when they use *Greedy* and *Random*, respectively. These results indicate that DOVE selects more effective OPs compared to *Greedy* and *Random*. As a result, in Fig. 3.6, the reduced cellular traffic of DOVE (or *DOVE*$^+$) is close to that of *Perfect*. Note that *Greedy* selects an RN without time consideration. Thus, a lot of consumers fail to retrieve a file when providers cannot store it timely in the OPs. Similarly, *Random* does not consider the travel times of vehicles and randomly selects OPs so that it also shows the low offloading ratio. The result of *Greedy* reveals that a lot of consumers fail to receive a file, as providers cannot store it timely in the OPs. Thus, *Greedy* and *Random* show the reduction of about a half of cellular traffic, compared to *Perfect* as shown in Fig. 3.6.

Fig. 3.7 shows how many request vehicles successfully perform offloading operations according to the TCC's schedule. Vehicles fail in offloading operations when consumers cannot obtain the file from the scheduled OP within the tolerance time or providers cannot store the file in the OP. As shown in Fig. 3.7, DOVE and *DOVE*$^+$ show the high success ratio (96% and 93%, respectively) while *Greedy* and *Random* exhibit the lower success ratio. Because many request vehicles using *Greedy* or *Random* may fail to retrieve a file due to the above two types of the failures. In DOVE, about 4% of request vehicles fail in offloading operations due to the error of travel time prediction. This incorrect prediction is caused by a variation of waiting time due to traffic signals. Compared to DOVE, *DOVE*$^+$ shows a little lower success ra-

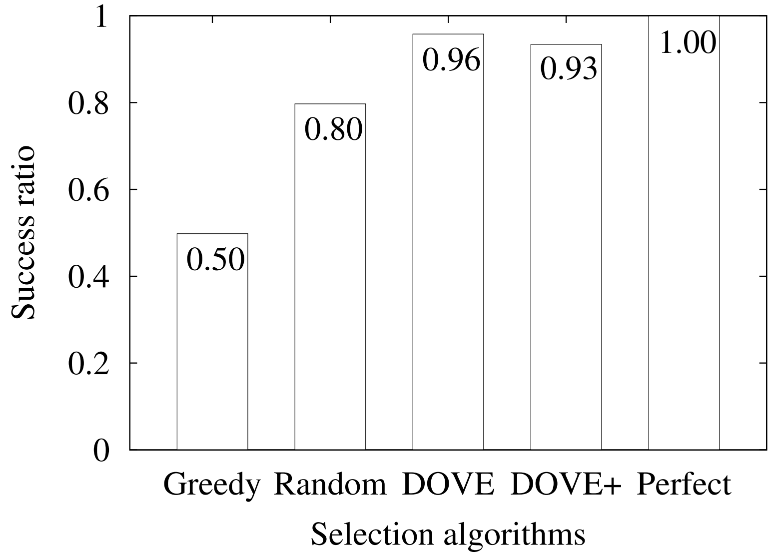


Fig. 3.7. The success ratio of offloading operations.

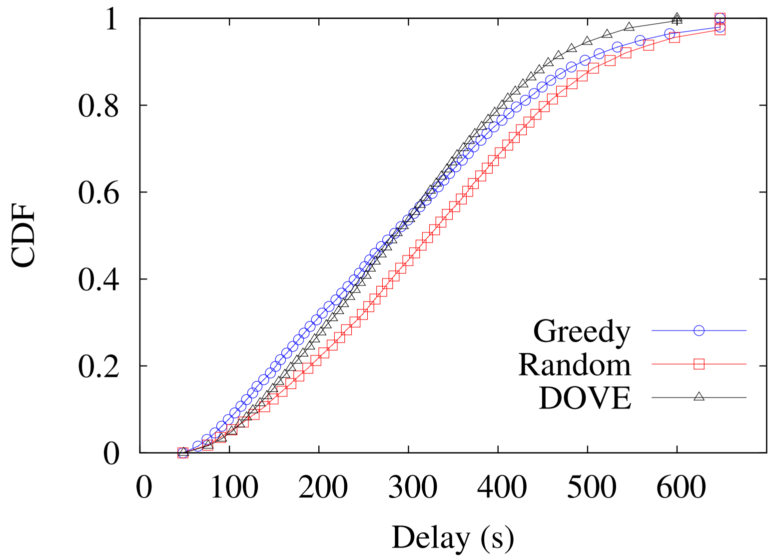


Fig. 3.8. The CDF of content retrieval delay of consumers.

tio (93%) since the prediction error of provider can cause more serious impact (i.e., higher failure ratio) on multiple donations. Note that about 50% of request vehicles using *Greedy* fail in offloading operations although *Greedy* chooses an RN that covers the largest number of request vehicles at each iteration. It shows the importance of travel time prediction. Thus, if TCC can perform better travel time prediction by real-time traffic measurement, our DOVE will provide a better data offloading.

Fig. 3.8 shows the cumulative distribution function (CDF) of content retrieval delays. We analyze the content retrieval delay of each consumer. Note that we omit the results of $DOVE^{+}$ and *Perfect* since they shows the similar performance to DOVE, and plotting their performance values hinders the readability of results. As shown in Fig. 3.8, all the consumers using DOVE obtain the file within the delay bound (i.e., 600 sec). However, *Greedy* and *Random* show the non-negligible portion of consumers that exceed the delay bound (3.2% and 4.3%, respectively). This is because *Greedy* and *Random* select OPs without considering the travel times of vehicles. Note that many vehicles using *Greedy* have the shorter content retrieval delay than DOVE. In DOVE, all vehicles are scheduled to utilize OPs considering the cellular downloading time γ (of a provider) to reduce the failure probability of offloading operations. However, *Greedy* selects OPs although travel times are less than the cellular downloading time, which leads to the fast content retrieval using cellular links due to the failure of offloading. As a result, about a half of consumers using *Greedy* show shorter content retrieval delay than vehicles using DOVE.

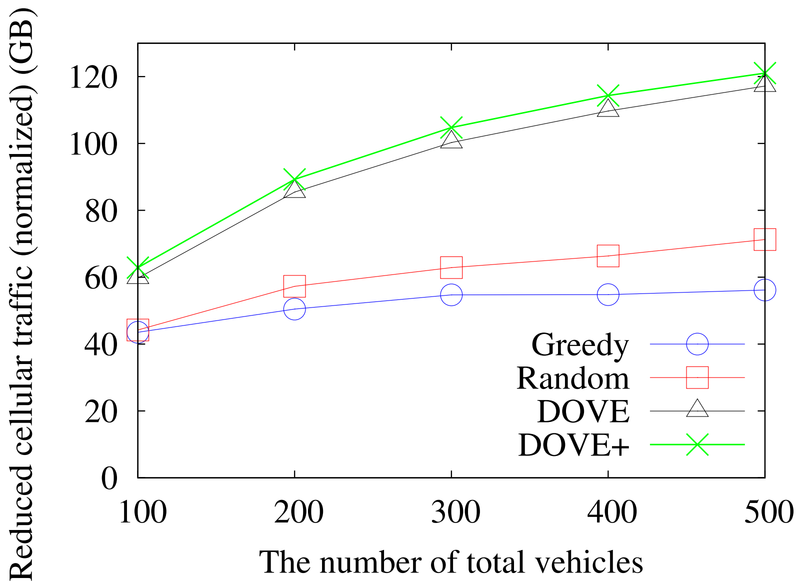


Fig. 3.9. The impact of vehicle number on the traffic reduction.

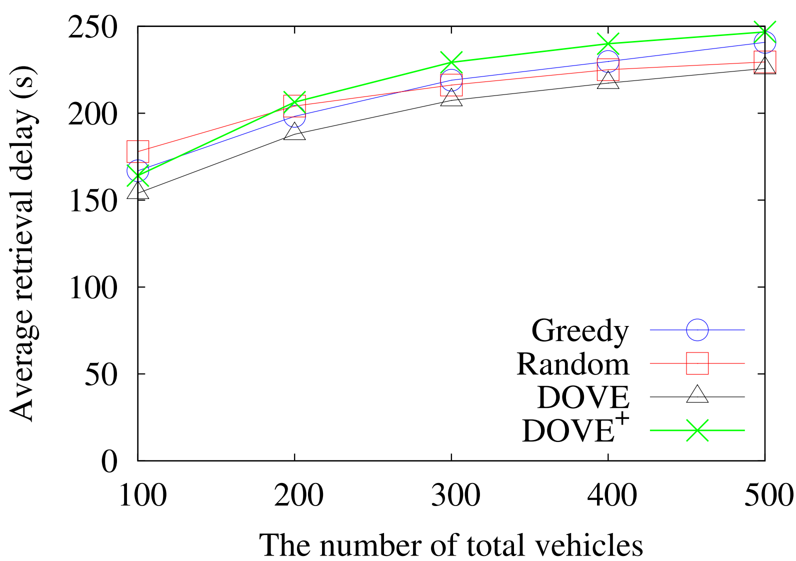


Fig. 3.10. The impact of vehicle number on the average retrieval delay.

3.5.2 The Impact of Vehicle Number

Assume that the number of DOVE users is proportional to the total vehicle number. Then, the number of vehicles in the road network determines the number of vehicles participating in DOVE service. In this chapter, we show how many vehicles can offload their traffic from cellular links to the vehicular network, as the number of DOVE users increases. In our simulations, we assume that 5% of vehicles participate in DOVE service to retrieve the same content file. As more vehicles in the road network request the same popular file, both the number of overlapped RNs (candidates for OPs) and the number of consumers passing through each OP also increase together. Thus, a significant reduction in cellular traffic is expected when vehicles utilize the OPs for data offloading.

Fig. 3.9 shows the reduction of cellular traffic with varying the number of vehicles, that is, from 100 to 500. For comparison, we normalize the results since the number of vehicles requesting the same file is changed under different vehicle numbers. In Fig. 3.9, as the number of vehicles increases, DOVE and *DOVE*$^+$ reduce a significant amount of cellular traffic. However, *Greedy* and *Random* show the marginal difference in traffic reduction. This is because many vehicles using *Greedy* and *Random* fail in offloading operations, leading to the performance degradation. In contrast, most vehicles using DOVE and *DOVE*$^+$ successfully perform offloading operations since they select OPs based on the time prediction. These results indicate the growth in consumers passing through each OP causes the performance degradation when providers fail to store the file at OPs.

To investigate the content retrieval delay of vehicles, we define the term *average retrieval delay* to be the average of the *content retrieval delays* of all vehicles. In

Fig. 3.10, we notice that the average retrieval delays of all the algorithms increase as the number of vehicles increases. This is because many vehicles postpone content downloading and try to retrieve the file via vehicular networks, which increases the content retrieval delay of each consumer, leading to the longer average retrieval delay. However, the content retrieval delay of a vehicle in *Greedy* and *Random* can be longer than the expected travel time to the selected OP when vehicles fail in offloading operations and switch to cellular links. Also, the average retrieval delay of *DOVE*$^+$ is longer than DOVE since multiple donations of *DOVE*$^+$ increase vehicles' travel time to the scheduled OPs.

3.5.3 The Impact of Vehicle Speed

We investigate how the change of mean vehicle speed affects the performance of data offloading. Fig. 3.11 shows the reduced cellular traffic under different mean vehicle speeds. As shown in the Fig. 3.11, for DOVE, *DOVE*$^+$, and *Random*, the higher vehicle speed somewhat increases the reduced cellular traffic. This is because the high vehicle speed yields the shorter travel time of vehicles, so this increases the number of vehicles that can utilize the OPs within the delay bound (i.e., 600 sec). However, the high vehicle speed slightly decreases the reduced cellular traffic of *Greedy*. This is because *Greedy* selects an RN that covers the largest number of vehicles, so the failure of providers increases the failure of consumers exploiting OPs.

In Fig. 3.12, the higher vehicle speed decreases the average retrieval delay of all the algorithms. This is because the shorter travel time of vehicles decreases the average retrieval delay. Compared to DOVE, the lower vehicle speed highly increases the average retrieval delay of *Greedy* and *Random*. This is because a lot of vehicles

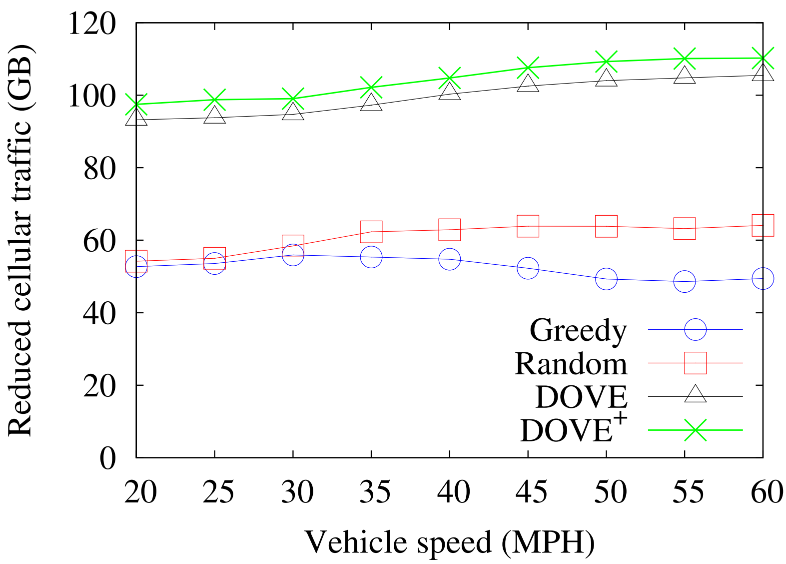


Fig. 3.11. The impact of vehicle speed on the traffic reduction.

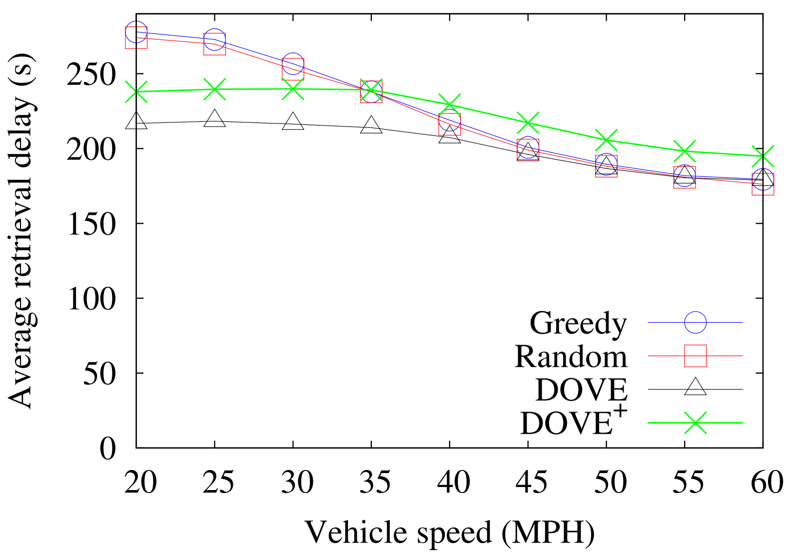


Fig. 3.12. The impact of vehicle speed on the average retrieval delay.

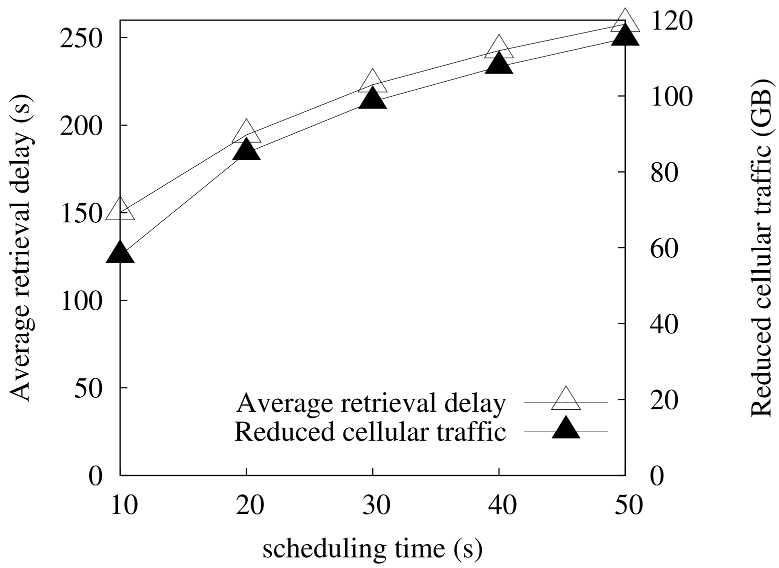


Fig. 3.13. The impact of scheduling time of the TCC.

cannot retrieve the file until the delay bound, leading to the increase of the content retrieval delay. Note that the average retrieval delay of *DOVE*$^{+}$ is longer than DOVE because of the longer travel time to the scheduled OPs, but the trends of two algorithms are the same.

3.5.4 The Impact of Waiting Time

We also investigate the impact of the waiting time of the TCC on the performance. To show the trend of performance, we assume the requests for the popular file from the request vehicles arrive at the TCC with the fixed rate of 0.5 requests per second. Fig. 3.13 shows the average retrieval delay and the reduced cellular traffic under different waiting times of the TCC. The longer waiting time leads to the more reduction of cellular traffic, but it increases the average retrieval delay. This is because the longer waiting time can increase the number of vehicles that request the same file with a high probability.

3.5.5 The Impact of Deployment Ratio and Tolerance Time

We investigate how the partial deployment of RNs and tolerance time affect the performance of DOVE. Fig. 3.14 shows the reduced cellular traffic under different deployment ratios, that is, from 10% to 100%. When the ratio is 100%, RNs are assumed to be deployed at all intersections. Note that we assume that RNs are deployed starting from the center toward the boundary of the road network. In all the algorithms, the higher deployment ratio leads to the less usage of cellular links. Compared to DOVE (or *DOVE*$^{+}$), *Greedy* and *Random* achieve lower performance gain since they cannot fully utilize the densely deployed RNs. Interestingly, DOVE shows that RNs in only 20% of intersections can reduce the 55.7 GB of cellular traffic, that is, about half of traffic reduction in the full deployment scenario (i.e., 100.3 GB). This result indicates our DOVE framework can achieve substantial data offloading even with the partial deployment scenario. In our simulation setting, the deployment ratio of 50% with DOVE algorithm shows the 10% of performance difference (91.5 GB) compared to the full deployment scenario.

Fig. 3.15 shows the reduced cellular traffic under different tolerance times, that is, from 300 sec to 900 sec. As shown in Fig. 3.15, DOVE and *DOVE*$^{+}$ show almost comparable performance, and the longer tolerance time increases the reduced cellular traffic in all the algorithms. This result implies that more tolerant users have more chances to offload their traffic from cellular links to RNs through DOVE. Also, longer tolerance time can increase the success probability of offloading operations since it

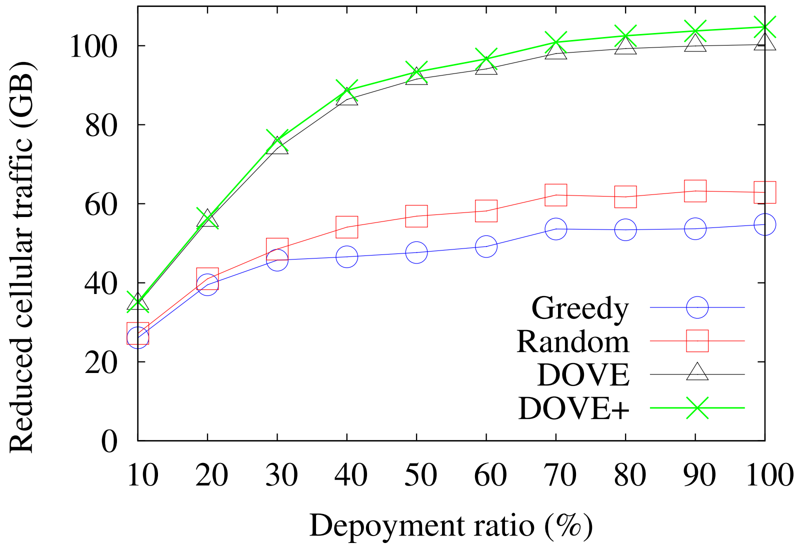


Fig. 3.14. The impact of partial deployment of RNs.

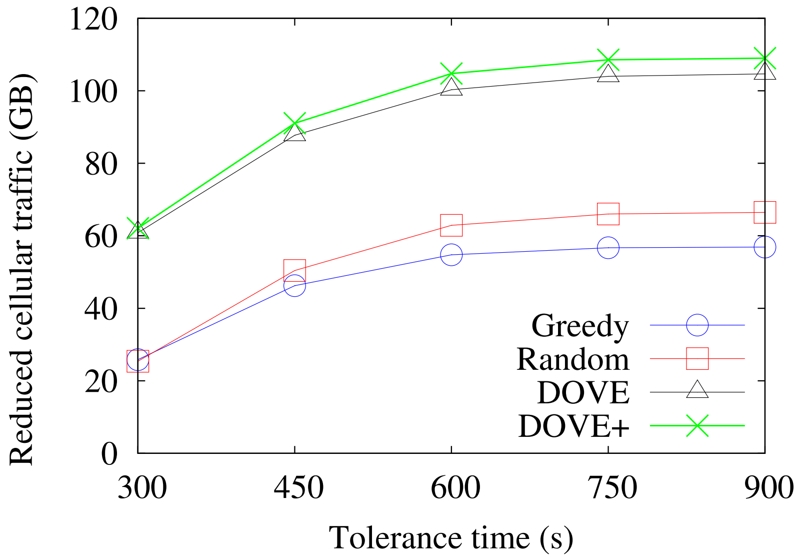


Fig. 3.15. The impact of tolerance time.

can reduce the effect of prediction error caused by a variation of waiting time due to traffic signals. Therefore, it can be concluded that DOVE can provide cost-effective offloading service with infrastructure nodes for the driving safety.

Chapter 4

Conclusion

This dissertation investigated the traffic reduction schemes to address the traffic explosion problem, which is becoming a serious concern for network providers. In particular, we focused on two different domains, information-centric networking and information-centric vehicular networks.

We first proposed a content discovery scheme called SCAN for information-centric networking. SCAN improves both reachability and efficiency by using a name-based routing and content discovery called scanning. When there are cached copies of a requested content in the vicinity, scanning is performed to find nearby cached contents to efficiently deliver content. In order to handle the huge number of cached contents and reduce the overhead of information exchange, SCAN exchanges the compressed content information using Bloom filters. Simulation results demonstrate that SCAN can deliver the contents to the users faster than other schemes since it can locate nearby cached contents. Furthermore, SCAN reduces the total volume of network traffic and provides better load balancing among the links. In the future, we will investigate forwarding strategies and parallel transmission mechanisms that utilize a multi-path to enhance the performance of SCAN.

We next proposed a Data Offloading framework through Vehicular nEtworks (DOVE). In DOVE, vehicle trajectories are utilized for data offloading. To select effective offloading positions (OPs), we formulated the problem of selecting offloading positions (OPs) as a spatio-temporal set-covering problem and then proposed a time-prediction based set-covering algorithm. Simulation results show that DOVE can reduce about 57% of cellular link usage via OPs. We believe our DOVE will be used as one of solutions to resolve the mobile traffic explosion. As future work, we will investigate how to use RSUs as providers that can fetch the data from the Internet without using cellular links, and also how to deal with large-size content files for data offloading with multiple OPs.

Bibliography

[1] "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013-2018," Feb. 2014.

[2] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," in *Proc. of ACM SIGCOMM*, 2007.

[3] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, kc claffy, D. Krioukov, D. Massey, C. Papadopolos, T. Abdelzaher, L. Wang, P. Crowley, and E. Yeh, "Named Data Networking (NDN) Project," Palo Alto Research Center (PARC), Tech. Rep., Oct. 2010.

[4] V. Jacobson, D. K. Smetter, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proc. of ACM CoNEXT*, 2009.

[5] K. Cho, H. Jung, M. Lee, D. Ko, T. T. Kwon, and Y. Choi, "How can an ISP merge with a CDN?" *IEEE Communications Magazine*, vol. 49, no. 10, pp. 156–162, Jun. 2011.

[6] K. Visala, D. Lagutin, and S. Tarkoma, "Lanes: An inter-domain data-oriented routing architecture," in *Proc. of the 2009 Workshop on Re-architecting the Internet*, 2009.

[7] C. Dannewitz, D. Kutscher, B. Ohlman, S. Farrell, B. Ahlgren, and H. Karl, "Network of information (netinf) - an information-centric networking architecture," *Elsevier Computer Communications*, vol. 36, no. 7, pp. 721–735, Apr. 2013.

[8] G. García-de-Blas, A. Beben, F. J. Ramón-Salguero, A. Maeso, I. Psaras, G. Pavlou, N. Wang, J. Sliwinski, S. Spirou, S. Soursos, and E. Hadjioannou, "COMET: content mediator architecture for content-aware networks," in *Proc. Future Network & Mobile Summit*, 2011.

[9] Z. Ming, M. Xu, and D. Wang, “Age-based Cooperative Caching in Information-Centric Networks,” in *Proc. of IEEE INFOCOM Workshop NOMEN*, 2012.

[10] I. Psaras, W. K. Chai, and G. Pavlou, “Probabilistic In-Network Caching for Information-Centric Networks,” in *Proc. of ACM SIGCOMM Workshop ICN*, 2012.

[11] K. Cho, M. Lee, K. Park, T. T. Kwon, Y. Choi, and S. Pack, “ WAVE: Popularity-based and Collaborative In-network Caching for Content-Oriented Networks,” in *Proc. of IEEE INFOCOM Workshop NOMEN*, 2012.

[12] W. K. Chai, D. He, I. Psaras, and G. Pavlou, “Cache Less for More in Information-Centric Networks,” in *Proc. of IFIP Networking*, 2012.

[13] M. Lee, J. Song, K. Cho, S. Pack, T. T. Kwon, J. Kangasharju, and Y. Choi, “Content discovery for information-centric networking,” *Elsevier Computer Networks*, Oct. 2014.

[14] Y. L. Morgan, “Notes on DSRC & WAVE Standards Suite: Its Architecture, Design, and Characteristics,” *IEEE Communications Surveys & Tutorials*, vol. 12, no. 4, pp. 504–518, Oct. 2010.

[15] J. Jeong, S. Guo, Y. Gu, T. He, and D. Du, “TSF: Trajectory-based Statistical Forwarding for Infrastructure-to-Vehicle Data Delivery in Vehicular Networks,” in *Proc. of IEEE ICDCS*, 2010.

[16] J. Chi, Y. Jo, H. Park, T. Hwang, and S. Park, “An Effective RSU Allocation Strategy for Maximizing Vehicular Network Connectivity,” *International Journal of Control and Automation*, vol. 6, no. 4, pp. 259–270, Aug. 2013.

[17] Paul Stenquist, “With Its Eye on Infotainment, General Motors Is Upgrading OnStar,” http://wheels.blogs.nytimes.com/2013/02/25/.

[18] Y. Li *et al.*, “Coding or Not: Optimal Mobile Data Offloading in Opportunistic Vehicular Networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 318–333, 2014.

[19] M. Lee, J. Song, J. P. Jeong, and T. T. Kwon, "Dove: Data offloading through spatio-temporal rendezvous in vehicular networks," in *Proc. of ICCCN*, 2015.

[20] R. Chiocchetti, D. Rossi, G. Rossini, G. Carofiglio, and D. Perino, "Exploit the Known or Explore the Unknown? Hamlet-Like Doubts in ICN," in *Proc. of ACM SIGCOMM Workshop ICN*, 2012.

[21] S. Eum, K. Nakauchi, M. Murata, Y. Shoji, and N. Nishinaga, "Potential based routing as a secondary best-effort routing for information centric networking (icn)," *Elsevier Computer Networks*, vol. 57, no. 16, pp. 3154–3164, Nov. 2013.

[22] M. Lee, K. Cho, K. Park, T. T. Kwon, and Y. Choi, "SCAN: Scalable Content Routing for Content-Aware Networking," in *Proc. of IEEE ICC*, 2011.

[23] Y. Wang, K. Lee, B. Venkataraman, R. L. Shamanna, I. Rhee, and S. Yang, "Advertising Cached Contents in the Control Plane: Necessity and Feasibility," in *Proc. of IEEE INFOCOM Workshop NOMEN*, 2012.

[24] L. Saino, I. Psaras, and G. Pavlou, "Hash-routing Schemes for Information-Centric Networking," in *Proc. of ACM SIGCOMM Workshop ICN*, 2013.

[25] R. Chiocchetti, D. Perino, G. Carofiglio, D. Rossi, and G. Rossini, "Inform: A dynamic interest forwarding mechanism for information centric networking," in *Proc. of ACM SIGCOMM Workshop ICN*, 2013.

[26] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang, "A case for stateful forwarding plane," *Elsevier Computer Communications*, vol. 36, no. 7, pp. 779–791, Apr. 2013.

[27] A. Kumar, J. Xu, and E. W. Zegura, "Efficient and scalable query routing for unstructured peer-to-peer networks," in *Proc. of IEEE INFOCOM*, 2005.

[28] S. C. Rhea and J. Kubiatowicz, "Probabilistic Location and Routing," in *Proc. of IEEE INFOCOM*, 2002.

[29] N. Hua, H. Song, and T. V. Lakshman, "Variable-stride multi-pattern matching for scalable deep packet inspection," in *Proc. of IEEE INFOCOM*, 2009.

[30] A. Broder and M. Mitzenmacher, "Network Apllications of Bloom Filters: A Survey," *Internet Mathematics*, vol. 1, no. 4, pp. 485–509, Jan. 2011.

[31] W. So, A. Narayanan, and D. Oran, "Named data networking on a router: Fast and dos-resistant forwarding with hash tables," in *Proc. of ANCS*, 2013.

[32] G. Rossini and D. Rossi, "Evaluating CCN multi-path interest forwarding strategies," *Elsevier Computer Communications*, vol. 36, no. 7, pp. 771–778, Apr. 2013.

[33] S. Arianfar, P. Nikander, and J. Ott, "On Content-Centric Router Design and Implications," in *Proc. of ACM SIGCOMM Workshop ReArch*, 2010.

[34] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, "How to model an internetwork," in *Proc. of IEEE INFOCOM*, 1996.

[35] Y. Wang, K. He, H. Dai, W. Meng, J. Jiang, B. Liu, and Y. Chen, "Scalable name lookup in ndn using effective name component encoding," in *Proc. of IEEE ICDCS*, 2012.

[36] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong, "Mobile data offloading: How much can WiFi deliver?" in *Proc. of ACM CoNEXT*, 2010.

[37] A. J. Nicholson and B. D. Noble, "BreadCrumbs: Forecasting mobile connectivity," in *Proc. of ACM MOBICOM*, 2008.

[38] Y. Im, C. Joe-Wong, S. Ha, S. Sen, T. Kwon, and M. Chiang, "Amuse: Empowering users for cost-aware offloading with throughput-delay tradeoffs," in *Proc. of IEEE INFOCOM*, 2013.

[39] S. Woo, E. Jeong, S. Park, J. Lee, S. Ihm, and K. Park, "Comparison of caching strategies in modern cellular backhaul networks," in *Proc. of ACM MobiSys*, 2013.

[40] PATH Project, "Investigating the Potential Benefits of Broadcasted Signal Phase and Timing (SPaT) Data under IntelliDrive," University of California, Berkeley, Tech. Rep., May 2011.

[41] J. Jeong, S. Guo, Y. Gu, T. He, and D. Du, “Trajectory-Based Data Forwarding for Light-Traffic Vehicular Ad-Hoc Networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 5, pp. 743–757, May 2011.

[42] I. Leontiadis and C. Mascolo, “GeOpps: Geographical Opportunistic Routing for Vehicular Networks,” in *Proc. of IEEE WoWMoM*, 2007.

[43] I. Leontiadis, P. Costa, and C. Mascolo, “Extending Access Point Connectivity through Opportunistic Routing in Vehicular Networks,” in *Proc. of IEEE INFOCOM*, 2010.

[44] I. Leontiadis, P. Costa, and C. Mascolo, “A Hybrid Approach for Content-based Publish/Subscribe in Vehicular Networks,” *Elsevier Pervasive and Mobile Computing*, vol. 5, no. 6, pp. 697–713, Dec. 2009.

[45] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter, “MDDV: A Mobility-Centric Data Dissemination Algorithm for Vehicular Networks,” in *Proc. of ACM VANET*, 2004.

[46] B. Han, P. Hui, V. A. Kumar, M. V. Marathe, G. Pei, and A. Srinivasan, “Cellular traffic offloading through opportunistic communications: a case study,” in *Proc. of ACM CHANTS*, 2010.

[47] Y. Li, G. Su, P. Hui, D. Jin, L. Su, and L. Zeng, “Multiple mobile data offloading through delay tolerant networks,” in *Proc. of ACM CHANTS*, 2011.

[48] X. Wang, M. Chen, Z. Han, T. Kwon, and Y. Choi, “Content dissemination by pushing and sharing in mobile cellular networks: An analytical study,” in *Proc. of IEEE MASS*, 2012.

[49] V. A. Siris and D. Kalyvas, “Enhancing mobile data offloading with mobility prediction and prefetching,” in *Proc. of ACM MobiArch workshop*, 2012.

[50] F. Malandrino *et al.*, “Offloading cellular networks through its content download,” in *Proc. of IEEE SECON*, 2012.

[51] Philadelphia Department of Transportation, "Traffic Control Center," http://philadelphia.pahighways.com/philadelphiatcc.html.

[52] Jupiter Research, "Municipal Wireless: Partner to Spread Risks and Costs While Maximizing Benefit Opportunities," Jupiter Research, Tech. Rep., Jun. 2005.

[53] Garmin Ltd., "Garmin Traffic," http://www8.garmin.com/traffic/.

[54] D. S. Berry and D. M. Belmont, "Distribution of Vehicle Speeds and Travel Times," in *Proc. of the Second Berkeley Symposium on Mathematical Statistics and Probability*, Jul. 1950.

[55] U. Feige, "A threshold of ln n for approximating set cover," *J. ACM*, vol. 45, no. 4, pp. 634–652, Jul. 1998.

[56] M. MacDougall, *Simulating Computer Systems: Techniques and Tools*. MIT Press, 1987.

[57] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," *Wireless Communications and Mobility Computing (WCMC): Special Issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, pp. 483–502, Aug. 2002.

[58] F. Bai, N. Sadagopan, and A. Helmy, "IMPORTANT: A framework to systematically analyze the Impact of Mobility on Performance of RouTing protocols for Adhoc NeTworks," in *INFOCOM*. IEEE, Mar. 2003.

[59] V. Muchuruza and R. Mussa, "Traffic Operation and Safety Analyses of Minimum Speed Limits on Florida Rural Interstate Highways," in *Proc. of MIDCON*, 2005.

초록

현재의 인터넷은 자원 공유를 목적으로 호스트간 통신 패러다임에 기반하여 설계되었지만, 오늘날 인터넷 사용 패턴은 콘텐트 획득에 집중되어있다. 이러한 이유로, 대부분의 인터넷 트래픽은 비디오 서비스나 P2P 파일 공유와 같은 콘텐트 획득에 의해 발생하고 있는 상황이다. 하지만, 현재 인터넷의 구조와 실제 사용 패턴의 괴리는 비효율적인 콘텐트 전달 (예, 동일한 인기있는 콘텐트에 대한 중복된 콘텐트 전송)을 야기하고 있고, 이는 트래픽 폭발 문제로 이어지고 있다. 이러한 이슈를 다루기 위해 (i) 인터넷 구조를 새롭게 설계하거나 (ii) 데이터 오프로딩 기법을 통해 네트워크 트래픽을 줄이려는 시도들이 있다. 본 학위 논문에서는 정보 중심 네트워킹과 정보 중심의 차량 네트워크라는 두가지 영역에서의 트래픽 감소 기법에 대해 탐구한다.

첫 번째로, 정보 중심 네트워킹을 위한 트래픽 감소 콘텐트 탐색 기법을 제안한다. 정보 중심 네트워킹은 트래픽 폭발과 같은 현재 인터넷의 문제점을 해결하기 위해, 초기 단계부터 인터넷 구조를 새롭게 설계하자는 방향으로 제안되었다. 정보 중심 네트워킹은 가까이에 존재하는 캐시된 콘텐트를 이용하거나 동일한 콘텐트 전송에 대한 중복된 전송을 줄이는 것을 통해 네트워크 트래픽 감소와 같은 이득을 제공할 수 있다. 하지만, 이전의 연구들은 이러한 캐시된 콘텐트를 이용하기 위해 기회주의적 캐시 일치 방식에 의존을 하고 있다. 이러한 방식은 콘텐트 소스로 가는 경로에 존재하는 캐시된 콘텐트만 이용할 수 있기 때문에 네트워크 곳곳에 있는 네트워크 내재 저장 공간을 충분히 이용하지 못하는 한계가 있다. 제안하는 기법인 SCAN은 네트워크에 산재된 캐시된 콘텐트를 더 잘 이용하기 위해서 가까이에 존재하는 캐시된 콘텐트를 탐색한

다. 이를 위해 SCAN은 블룸 필터를 사용하여 주변 라우터들 사이에서 캐시된 콘텐트에 대한 정보를 교환한다. 시뮬레이션을 통해 SCAN은 기회주의적 캐시 일치 방식의 기법에 비해 평균 홉 거리, 트래픽 양, 링크간 로드 분배에서 더 나은 성능을 보임을 알 수 있다.

다음으로, 정보 중심의 차량 네트워크를 위한 트래픽 오프로딩 기법을 제안한다. 무선 환경에서 급증하고 있는 모바일 트래픽은 모바일 네트워크 제공자에게 심각한 우려가 되고 있다. 이러한 트래픽 폭발 문제를 다루기 위해, 트래픽을 셀룰러 네트워크에서 WiFi 핫스팟이나 펨토셀과 같은 다른 네트워크로 오프로딩하려는 연구들이 있었다. 본 연구에서는 기존의 시도에서 더 나아가서 데이터 오프로딩을 위한 차량 네트워크의 잠재적 장점에 집중하여 차량 네트워크를 이용한 데이터 오프로딩 프레임워크인 DOVE를 제안한다. 제안하는 데이터 오프로딩 프레임워크는 차량 내 데이터 서비스를 위해 필요한 셀룰러 트래픽을 비용 효과가 높은 방식으로 감소시킬 수 있다. DOVE에서는 오프로딩을 목적으로 차량 이동 경로를 이용하고, 경제적인 비용 절감을 목적으로 차량에서 요청되는 콘텐트 파일들은 셀룰러 네트워크 대신 차량 네트워크를 통해 전달된다. 이를 위해 오프로딩 위치를 선택하는 문제를 시공간적 집합 덮개 문제로 만들고, 차량 이동 경로를 이용한 시간 예측 기반의 집합 덮개 알고리즘을 제안한다. 시뮬레이션 결과에 따르면, DOVE 프레임워크는 차량 네트워크를 통한 오프로딩을 수행하여 57%의 셀룰러 링크 사용량을 크게 감소시킬 수 있다.

주요어: 정보 중심 네트워킹, 정보 중심의 차량 네트워크, 트래픽 감소, 콘텐트 탐색, 데이터 오프로딩

학번: 2007-21039